



User Guide

July 2014

Version i19.91

© 2011-2014 Sumo Logic, Inc. All rights reserved. Sumo Logic, the Sumo Logic logo, and any other product or service names or slogans contained in this document are trademarks of Sumo Logic and its suppliers or licensors, and may not be copied, imitated or used, in whole or in part, without the prior written permission of Sumo Logic or the applicable trademark holder. All other trademarks, registered trademarks, product names and company names or logos mentioned in this document are the property of their respective owners. Reference to any products, services, processes or other information, by trade name, trademark, manufacturer, supplier or otherwise does not constitute or imply endorsement, sponsorship or recommendation thereof by us.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Sumo Logic.

Sumo Logic may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Sumo Logic, the furnishing of this document does not give you any license to these patents, trademarks copyrights, or other intellectual property.

The information in this document is subject to change without notice. Sumo Logic shall not be liable for any damages resulting from technical errors or omissions which may be present in this document, or from use of this document.

Contents

Sumo Logic Overview	17
Sumo Logic Account Types	18
Sumo Logic components	21
Sumo Logic Collectors and Sources	21
Sumo Logic Cloud	21
Sumo Logic Web Application	21
Sumo Logic Source Types	22
Installed Collector Sources	22
Hosted Collector Sources	22
Designing your deployment	23
Installed Collectors	23
Hosted Collectors	23
A note about logging levels	24
Download and Install Collectors	25
What's the difference between Collector types?	26
Installed Collectors	27
Installed Collector Requirements	28
How can I check for memory issues with a Collector?	28
Single Installed Collector configuration	29
Multiple Installed Collector configuration	29
Cloud or data center deployment	30
Configuring Sources	30
Collecting Local Files	31
Collecting Remote Files	31
Collecting from Syslog Sources	31
Collecting Windows Events	32
Collecting from a Script	32
Linux (64 bit) Collectors	32
	33
Windows 2008 (64 bit) Collectors	33
Downloading Installed Collector Software	34
Installing a Collector	35
Step 1. Download the tarball	39
Step 2. Configure Sources	39
Step 3. Modify wrapper.conf	40
Step 4. Create sumo.conf	40
Step 5. Run the Collector	41
About updating your configuration	41
Step 1. Download the RPM package	41
Step 2. Configure Sources	42
Step 3. Configure sumo.conf	42

Step 4. Install the Collector	42
Step 1. Download the Collector	43
Step 2. Configure sumo.conf	43
Step 3. Configure Sources using the Collector Management API	44
Step 4. Install the Collector	44
Step 1. Download the Collector	45
Step 2. Configure Sources	45
Step 3. Configure sumo.conf	45
Step 4. Install the Collector	46
Step 1. Download the tarball	46
Step 2. Configure Sources	47
Step 3. Modify wrapper.conf	47
Step 4. Create sumo.conf	48
Step 5. Run the Collector	48
Step 1. Download the Collector	49
Step 2. Configure Sources	49
Step 3. Configure sumo.conf	49
Step 4. Install the Collector	50
Step 1. Download the tarball	50
Step 2. Configure Sources	51
Step 3. Modify wrapper.conf	51
Step 4. Create sumo.conf	52
Step 5. Run the Collector	52
Advanced Collector installation	53
A note about JRE	53
sumo.conf parameters	53
Creating sumo.conf	54
Using sumo.conf to disable Collector options	55
Disabling remote Collector upgrades	55
Disabling Source types	55
Forcing a Collector's Name with Clobber	55
Setting a Collector as Ephemeral	56
What happens to logs collected from ephemeral Collectors?	57
Connecting via an authentication proxy	57
Connecting via an NTLM proxy	58
Hosted Collectors	59
Setting up a Hosted Collector	59
Step 3. Add Sources	62
Configuring a Local File Source	63
How does Sumo Logic handle log file rotation?	65
Configuring a Remote File Source	66
Configuring a Local Windows Events Source	69

Configuring a Remote Windows Event Log Source	71
Prerequisites for Windows Collections	73
Step 1. Install a Sumo Logic Collector.	73
Step 2. Set UNC share permissions.	73
Configuring a Local Windows Performance Monitor Log Source	76
Configuring a Remote Windows Performance Monitor Log Source	80
Configuring a Syslog Source	83
Configuring a Script Source	85
When should I set a timeout for my script?	87
Troubleshooting Script Source Issues	88
Advanced Topic: Using CRON Expressions	88
Configuring a Script Action	93
Step 1. Set up the Script Action.	93
Step 2. Set up a Scheduled Search.	94
About the search results file	95
Adding an Amazon S3 Source	97
Configuring an Amazon S3 Source	97
Before you begin	97
About Path Expressions in S3 Sources	97
Granting Access to an S3 bucket	99
Policy JSON	102
About setting the S3 Scan Interval	103
Configuring an HTTP Source	105
Adding an HTTP Source	105
Uploading data to an HTTP Source	107
Generating a new URL	108
Setting Source timestamp options	110
Specifying a timestamp format	110
Supported Timestamp Conventions	111
Are Unix timestamps supported?	112
Timestamp examples	112
Using JSON to Configure Sources	113
Non-configurable parameters	113
Time zone format	113
Generic parameters	113
Local File Source	114
Remote File Source	116
Local Windows Event Log Source	118
Remote Windows Event Log Source	120
Syslog Source	122
Script Source	124
Using Wildcards in Paths	127

Specifying Paths to collect from	127
Using Wildcards in the Blacklist Field	127
Using the Sumo Logic Web Application	128
Supported Browsers	129
Understanding the Web Application user interface	130
Welcome	130
Search	130
Status	131
Collectors	131
Users	132
Account	132
Preferences	133
Navigating through Messages in the Search Tab	135
Preferences page	136
Editing your Profile	136
Creating and Managing Access Keys	136
Editing your Preferences	136
Changing your password	138
Web Application keyboard shortcuts	139
Search page keyboard shortcuts	139
Searching and Analyzing	140
Running a basic search	141
Keyword search expression	142
Sumo Logic operators	142
Search Highlighting	143
Wildcards in Full Text Searches	144
Time Range Expressions	145
Saving a Search	147
Sharing a Search Link	148
Setting the Time Range of a Search	149
Using Receipt Time	150
Parsing and Naming a Field	152
Pausing or Cancelling a Search	154
Modifying a Search from the Messages Tab	154
What's appended to my original search?	155
Searching with LogReduce	157
Metadata Searches	157
What metadata fields can I search?	157
Searching Surrounding Messages	159
Exporting Search Results	160
Graphing Search Results	161

Setting Messages tab preferences	162
Writing Efficient Queries	164
Parsing	165
Parse Operator	165
Parse Regex or Extract Operator	166
Extracting multiple values for a single field	167
Parse nodrop Option	168
Extracting Fields with Parser Libraries	169
Parsing Apache Logs	171
Parsing Cisco ASA Logs	174
Parsing Microsoft IIS Logs	175
Windows 2008 Parser	176
Parsing XML	178
Group Operator and Group-By Functions	181
Group Operator	181
avg	182
count, count_distinct, and count_frequent	182
first and last	184
min and max	187
most_recent and least_recent	187
pct	189
stddev	190
sum	190
Sumo Logic Operators and Expressions	193
Search Syntax Overview	193
Refining a Query	194
Conditional Expressions	196
Using the "not" option	197
Use where to check for null values	197
Accum operator	198
CIDR Operator	200
Concat Operator	201
CSV Operator	203
Difference (Diff) Operator	204
Fields Operator	206
Format Operator	208
Mapping IP addresses with the geo lookup operator	209
IPv4ToNumber Operator	214
isNull Operator	215
Join Operator	217
Examples	218
JSON Operator	220

Keyvalue operator	222
Inference mode syntax	223
Regular Expression mode syntax	223
Abbreviated syntax	224
Limit Operator	224
Lookup Operator	226
Luhn Operator (Credit Card Validator)	229
Matches Operator	231
Examples	231
Math Expressions	233
abs	233
round	234
ceil	234
floor	234
max	234
min	235
sqrt	235
cbrt	235
exp	235
expm1	235
log	236
log10	236
log1p	236
sin	236
cos	237
tan	237
asin	237
acos	237
atan	237
atan2	237
sinh	238
cosh	238
tanh	238
hypot	238
signum	238
toDegrees	239
toRadians	239
Rollingstd Operator	239
Save Operator	240
Using the Sessionize Operator	242
Smooth Operator	244
Sort Operator	246

Split Operator	247
Summarize Operator	249
Will my summarize search results match my keyword search results?	250
Running a Summarize query	250
Investigating the Others signature	252
Promoting or Demoting a Summarize Signature	253
Splitting a signature	254
What do the Relevance values mean?	258
Changing the display of Summarize results	258
Understanding Summarize Delta results	259
Options	259
Extending Summarize Delta with the Where Operator	260
Field Descriptions	260
Timeslice Operator	261
toLowerCase and toUpperCase Operators	263
Top Operator	264
Total operator	265
Trace Operator	267
Transpose Operator	269
URL Decode Operator	272
Sumo Logic Suggested Searches	274
Searches for the Apache Access Parser	274
Searches for the Apache Errors Parser	277
Searches for the Cisco ASA Parser	278
Suggested Searches for Linux OS Systems	279
Searches for the Microsoft IIS Parser	282
Searches for Windows 2008 Events	283
Search Optimization	286
Does all data need to be indexed?	287
Is there such a thing as creating too many indexes?	287
Choosing the right search optimization tool	288
Partitions	289
How are Partitions different from Sumo Logic Indices and Scheduled Views?	289
Running a Search Against a Partition	289
Running search across multiple Partitions	290
Using Partitions with Monitors and Dashboards	290
Managing Partitions	291
Adding a Partition	292
How do new Partitions affect your current system?	293
About Scheduled Views	294
How data is added to a Scheduled View	294
How are Scheduled Views different than Partitions and Sumo Logic Indices?	294

Designing Scheduled Views	294
Creating a Scheduled View	295
Running a Search Against a Scheduled View	296
Using Timeslices in Scheduled Views	297
Running a Search Against a Scheduled View	297
Data Volume Index	298
Enable the Data Volume Index	298
Using the Data Volume Index	298
Data Volume Index Message Format	299
Examples	300
Sumo Logic App for Data Volume	301
About Dashboards	302
What else do I need to know before I get started?	303
Using Dashboards in Sumo Logic Free Accounts	304
Adding Monitors to a Dashboard	305
Why does the data look different in the new Monitor?	306
Restricted Operators	306
Publishing Dashboards	307
Publishing Dashboards from the Library	307
Publishing from the Dashboards page	308
Launching a search from a Monitor	309
Zooming in on a Monitor	311
Pausing a Monitor	312
Editing Dashboards and Monitors	315
Changing Monitor chart properties	315
Changing the axis of a Monitor	316
Changing the name of a Dashboard or Monitor	316
Changing the time range of a Monitor	317
Changing the color of a bar or column by value range	318
Changing the type of a Monitor	319
Saving edits to a Monitor	319
Creating combo chart Monitors	320
Creating Box Plot Charts	322
Creating Single Value Charts	324
Numerical Single Value Chart	324
String Single Value Chart	327
Boolean Single Value Chart	328
Setting Firefox Permissions to View Dashboards	331
Using the Library	332
How the Library works	333

Saving a Search	335
Scheduling searches	337
Canceling or Editing Scheduled Searches	339
Running a Search from an Alert Email	340
Setting up Real Time Alerts	341
Publishing a search from the Library	345
Publishing Dashboards	347
Publishing Dashboards from the Library	347
Publishing from the Dashboards page	348
Subscribing to searches and Dashboards	349
Copying Content	350
Installing Apps from the Library	352
Using Library keyboard shortcuts	354
Sumo Logic Applications	355
Sumo Logic Log Analysis QuickStart Application	356
Sumo Logic Log Analysis QuickStart App Dashboards	356
Visits Dashboard	356
Keywords and Metadata Dashboard	357
Collectors and Source Monitoring Dashboard	359
Installing and starting the Log Analysis QuickStart App	360
Sumo Logic App for Data Volume	362
Sumo Logic App for Data Volume Dashboards	362
Installing the Data Volume App	364
Sumo Logic App for Apache	366
Log Types	366
Sumo Logic App for Apache Dashboards	366
Installing the Sumo Logic App for Apache	370
Sumo Logic App for AWS CloudTrail	373
Before you begin	373
Using the App for CloudTrail in multiple environments	373
Sumo Logic App for AWS CloudTrail Dashboards	373
What if data isn't displaying in all Monitors?	373
Collecting logs for the Sumo Logic App for AWS CloudTrail	376
Installing and starting the AWS CloudTrail App	378
Configuring the App for AWS CloudTrail in multiple environments	380
Sumo Logic App for AWS CloudFront	381
Visitor Statistics Dashboard	381
Web Operations Dashboard	382
Sumo Logic App for AWS Elastic Load Balancing	383
Collecting logs for the AWS Elastic Load Balancing App	383
AWS Elastic Load Balancing Dashboards	384

What if data isn't displaying in all Monitors?	385
Sumo Logic App for Cisco	388
Log Types	388
Sumo Logic App for Cisco Dashboard	388
Installing the Cisco App	390
Sumo Logic App for IIS	392
Log Types	392
Sumo Logic App for IIS Dashboards	392
Installing the IIS App	396
Sumo Logic App for Linux	399
Log Types	399
Sumo Logic App for Linux	399
Installing the Linux App	403
Sumo Logic App for Nginx	405
Log Types	405
Sumo Logic App for Nginx Dashboards	405
Installing the Nginx App	411
Sumo Logic App for Palo Alto Networks (PAN)	413
Log Types	413
Sumo Logic App for Palo Alto Networks (PAN) Dashboards	413
Installing the PAN App	416
Sumo Logic App for Varnish	419
Sumo Logic App for Varnish Dashboards	419
Installing the Varnish App	425
Sumo Logic App for VMware	427
Collecting logs for the Sumo Logic Application for VMware	427
Step 1: Install vMA	427
Step 2: Download and Install the Collector on vMA	428
Step 1: Configure a Syslog Source for the Collector.	429
Step 2: Configure Logs to be Collected	430
Step 1: Configure a Local File Source.	430
Step 2: Configure Performance Logs for Collection	431
Troubleshooting and Manual Testing	432
Collecting Historical Events	433
Installing the app for VMware	434
Sumo Logic App for Windows	437
Log Types	437
Sumo Logic App for Windows	437
Installing the Sumo Logic App for Windows	440
Sumo Logic App for Windows Active Directory	442
Collecting Active Directory log files	442

Verifying Active Directory Module	442
Installing the AD App	445
Sumo Logic App for Active Directory Dashboards	447
Understanding Source Mapping in App Installation	451
How can I change the sourceCategory associated with an app?	451
Requesting Apps	452
Sumo Logic Administration	453
Managing Installed Collectors and Sources	454
Upgrading Collectors using the Web Application	454
How will I know when an upgrade is available?	454
Troubleshooting upgrade failures	455
Searching for a Collector or Source	455
Editing a Collector	456
Editing a Source	457
Starting or stopping a Sumo Logic Collector	458
Filtering Source Data	459
Editing filters	465
Establishing Metadata Conventions	465
Using the Status page	468
Changing the Scale or Timeframe for a Collector	468
Viewing Stopped Collectors	469
Account page	470
Account Overview	470
Data Management tab	471
Before you begin	472
Set up Data Forwarding	472
Editing Data Forwarding Settings	473
Granting S3 permissions	474
Policy JSON	477
Managing Access Keys	478
Managing Usage Reports	479
Data Usage	479
Managing Data Volume	480
Monitor average daily volume	480
Use filters to limit the data volume	480
Using Role-Based Access Control with Sumo Logic	481
Managing Users and Roles	481
Managing User Roles	482
Administrator	482
Analyst	483
How do roles work together?	483

What about shared content?	484
Denying access to data	485
Granting access to data	486
Defining access based on metadata	486
Defining access based on records	487
Managing Users	489
Editing a user	490
Deactivating a user	491
Deleting a user	491
Managing Security Settings	494
Setting password policies	494
Where can I view my organization's current Sumo Logic password policy?	495
Whitelisting IP or CIDR addresses	496
Using Access Keys	498
Enabling a Support Account	501
Who can access my Support Account?	501
Do I need to create a special user account?	501
Managing Billing	503
Upgrading your Account	503
Who can upgrade my organization's account?	503
Can I upgrade more than once?	503
Using the Billing Page	505
Do other account types have Account Owners?	505
Managing Ingestion	508
Throttling	508
Account caps	508
Provisioning SAML	509
Prerequisites	509
Configuring SAML	509
Optional SAML features	512
Viewing SAML debug information	513
About Anomaly Detection	515
Who can use Anomaly Detection?	516
About the Anomalies Page	517
Adding Reports to the Anomalies Page	518
About Events and Incidents	519
Why are Events shown in different colors?	519
Naming and Labeling Events	520
Drilling Down into Events	521
Viewing Historical Incidents	522
Assigning Anomaly Detection Permissions to Users	524

Sumo Logic Beta Features **525**

 About Field Extraction 526

 Creating a Field Extraction Rule 526

 Running a Search Against Extracted Fields 527

 Using Lookup to Access Saved Data 528

CHAPTER 1

Sumo Logic Overview

The Sumo Logic solution provides everything you need to conduct real time forensics and log management for all of your IT data—without having to perform complex installations or upgrades, and without the need to manage and scale any hardware or storage. With fully elastic scalability, Sumo Logic is a fit for any size deployment.

In this section, we'll answer these questions:

- [What components make up a Sumo Logic solution?](#)
- [What is a Source?](#)
- [How many Collectors do I need?](#)
- [Where do I install the Collectors?](#)

If you need more information, [contact Sumo Logic Customer Support](#). They can assist you with specific configuration issues or help you to visualize and map large deployments.

Sumo Logic Account Types

There are three types of Sumo Logic accounts:

- **Free** accounts allow you to use the Sumo Logic tools available to Professional account holders, with a daily data volume limit of 500MB. Free accounts allow three users, with seven days of data retention. Free account holders can upgrade to Professional using the Sumo Logic Web Application.
- **Professional** accounts scale to meet your growing needs for user licenses, data retention, and volume options based on subscription. You can upgrade from a Professional to an Enterprise account at any time.
- **Enterprise** accounts, the premier Sumo Logic log management solution, are built to fit your organization's needs for data volume, data retention, and user management requirements. Enterprise accounts include the additional features of SAML-based SSO and Anomaly Detection.

Feature	Description	Free Account	Professional Account	Enterprise Account
Data volume	Daily amount of data sent to the Sumo Logic Cloud.	500MB per day	Based on subscription	Based on subscription
Data retention	Amount of time log data are stored in the Sumo Logic Cloud.	Seven days	Based on subscription	Based on subscription
LogReduce Analytics	Use a Sumo Logic algorithms to reduce log lines into manageable patterns.	✓	✓	✓
Search	Search historical and near real time data.	✓	✓	✓
Library	Save a search or a dashboard and <u>share it</u> with everyone in your Sumo Logic account.	✓	✓	✓
Data Collection	Collect data from one or more Sources (from both Local and Hosted Collectors).	✓	✓	✓
Data Forwarding	Forward your logs to an S3 Bucket after Sumo Logic has ingested it.	Upgrade required	✓	✓
Data Volume Index	Provides information on data volume usage.	✓	✓	✓
Applications	Install out of the box applications from Sumo Logic into your account.	Quickstart App only	✓	✓
Users	Administrators can set user privileges based on role access controls.	Three users	Based on subscription	Based on subscription

Multi-site Aggregation	<u>Collect</u> data across multiple infrastructures or environments.			
High Availability	Built-in high availability service infrastructure.			
Support	Access to product documentation and <u>Sumo Logic technical support</u> .			
SAML	Provision SAML-based SSO for your organization.	Upgrade required	Upgrade required	
Anomaly Detection	Machine learning and logic proactively detects abnormalities in your environment.	Upgrade required	Upgrade required	
<u>Collector Management API</u>	Configure multiple Collectors and Sources using a script, without needing to use the Sumo Logic Web Application UI.			
<u>Search API</u> and <u>Search Job API</u>	Access resources and log data within Sumo Logic by third party scripts and applications.	Upgrade required	Upgrade required	
<u>Dashboard API</u>	Access data produced by existing Monitors in your organization's Dashboards.	Upgrade required	Upgrade required	

Important notes on Sumo Logic Free accounts

Using a Free account is a great way to get to know Sumo Logic. While you're trying the Sumo Logic service, there are a few points that are important to be aware of:

- Free accounts run on seven day intervals. This means that over the course of seven days, you can't upload more than a total of 3.5GB of log data.
- If you begin to reach the 500MB daily limit, Sumo Logic sends an email to let you know. You can take action to reduce the amount of data you're uploading in order to stay below the limit.
- If the 500MB limit is surpassed, you'll receive an email letting you know that data in the Sumo Logic Cloud can no longer be searched (but additional data is still collected). However, if the data limit is fully exceeded, data collection stops (in addition to search being disabled). Disabled features will be available after your usage falls below 3.5GB when averaged over seven days (this could take one day, or up to seven days, depending on the amount of data you've uploaded and where you've uploaded it).
- In extreme situations, Free accounts may be disabled if the data volume continues to exceed the limits.
- Free accounts are limited to 10 dashboard monitors.
- For applications, Free accounts are limited to install the Log Analysis QuickStart app.
- The limitations of a Free account can't be changed, but you can upgrade to a Professional account at any time.



For Free and Professional accounts, it's important to keep track of your daily usage. For tips on how to monitor and limit the data you're sending to Sumo Logic, see [Managing Data Volume](#).

Sumo Logic components

A Sumo Logic solution is comprised of just a few components: Sumo Logic Collectors and Sources, the Sumo Logic Cloud, and the Sumo Logic Web Application. A Collector is a small application that gathers log data from your servers and sends it to the Sumo Logic Cloud. Using the Sumo Logic Web Application, you can interact with and analyze your data in the cloud in real time.

Sumo Logic Collectors and Sources

Sumo Logic Installed Collectors receive data from one or more Sources. Collectors collect raw log data, compress it, encrypt it, and send it to the Sumo Cloud, in real time. A single Sumo Logic Collector can collect up to 15,000 events per second or more and has fault tolerance during network or service outages. If you'd like to collect non-traditional machine data, a Script Source or Script Action provide a great deal of flexibility to collect files.

Sumo Logic Hosted Collectors collect data uploaded from Amazon S3 Sources, allowing you to move objects from your organization's S3 bucket directly into Sumo Logic.

Sumo Logic Cloud

The Sumo Logic Cloud is a secure, scalable repository for all of your operations, security, compliance, development, and other log data. The Sumo Logic Cloud stores, indexes, parses, and analyzes data, and provides unlimited horsepower with elastic scalability.

Sumo Logic Web Application

The Sumo Logic Web Application allows you to view and analyze your log data in the cloud. With a powerful and intuitive search capability, you can use the web application to expedite functions like forensic analysis, troubleshooting, and system health checks. The Sumo Logic Web Application provides access from anywhere since it is fully browser based. It also provides all required administration tools for managing your installation. This includes tools for administration, checking system status, managing your deployment, and direct access to download and activate Collectors.

Sumo Logic Source Types

There are several different types of Sources that can be configured for Installed and Hosted Collectors.

Installed Collector Sources

Depending on the type of data you need to collect, you can choose from the following:

Local File Source. To collect log messages from the same machine where a Collector is installed, create a Local File Source.

Remote File Source. Supported using SSH (Secure Shell), for remote file collections, you can install the Sumo Logic Collector on a network host that has network connectivity to all the remote hosts from which you wish to collect logs.

Syslog Source. Operates like a syslog server listening on the designated port to receive syslog messages. Set your syslog-enabled devices to send syslog data to the same port you specify in Sumo Logic Syslog Source configuration.

Local Windows Events Source. Collects local events you would normally see in the Windows Event Viewer. Setting up a Local Windows Event Source is a quick process. There are no prerequisites for setting up the Source, and you'll begin collecting logs within a minute or so.

Remote Windows Events Source. Collects the unique formats of Windows Events using the WMI (Windows Management Instrumentation) interface. Before collecting Windows Events from a remote machine, you'll need to configure a few settings.

Script Source. If you need to collect data that isn't stored in log files (like system performance metrics, database records, or perhaps data output from third-party monitoring solutions) you can use a Script Source that uses a script to fetch those custom sources of data. The script executes at defined intervals and then sends the data to Sumo Logic for analysis.

Hosted Collector Sources

Amazon S3 Source. Data stored in your organization's S3 bucket can be uploaded into Sumo Logic, allowing you to perform the same searches and forensics on data no matter where it's stored.

HTTP Source. An endpoint for receiving a file (or a batch of files) uploaded via a unique URL generated for the Source. The URL securely encodes the Collector and Source information. You can add as many HTTP Sources as you'd like to a single Hosted Collector.

Designing your deployment

Sumo Logic Installed Collectors are lightweight and efficient. You can choose to install a small number of Collectors to minimize maintenance or just because you want to keep your topology simple. Alternatively, you can choose to install many Collectors on many machines to distribute the bandwidth impact across your network rather than having it centralized. In addition, you'll want to consider Sumo Logic Hosted Collectors, which reside in the Cloud, allowing for seamless collection from Amazon S3 buckets and HTTP Sources.



What's the difference between Installed Collectors and Hosted Collectors? Learn more [here](#).

Installed Collectors

Installed Collectors are deployed in your environment, either on a local machine, a machine in your organization, or even an AMI. Installed Collectors require a software download and installation. Upgrades to Collector software are released regularly by Sumo Logic.

Consider having an Installed Collector on a dedicated machine if:

- You are running a very high-bandwidth network with high logging levels.
- You want a central collection point for many Sources.

Consider having more than one Installed Collector if:

- You expect the combined number of files coming into one Collector to exceed 500.
- Your hardware has memory or CPU limitations.
- You expect combined logging traffic for one Collector to be higher than 15,000 events per second.
- Your network clusters or regions are geographically separated.
- You prefer to install many Collectors, for example, one per machine to collect local files.

Sumo Logic Installed Collectors run on these operating systems:

- Linux 32-bit (x86/i386)
- Linux 64-bit (x86_64/amd64)
- Solaris (x86)
- Mac OS X
- Windows (32 bit/64 bit)

Hosted Collectors

Unlike Installed Collectors, Hosted Collectors don't require installation or activation, nor do Hosted Collectors have physical requirements, since they're hosted in AWS.

Because there are no performance issues to consider, you can configure as many S3 and HTTP Sources as you'd like for a single Hosted Collector. Consider setting up more than one Hosted Collector if you'd like to tag different data types with different metadata.

A note about logging levels

The more sensitive the logging level settings are for your applications and devices, the more logs will be sent to the Sumo Logic Cloud. In order to maximize the value of your log collection and analysis, set the logging level as high as you can without negatively impacting the CPU utilization of the machine where the Collector is running. The more searchable data you collect, the more information you have for forensic analysis and troubleshooting.

If you have additional questions, a [Sumo Logic sales representative](#) can help determine specific recommendations for your installation.



CHAPTER 1

Download and Install Collectors

The Sumo Logic solution provides everything you need to conduct real time forensics and log management for all of your IT data—without having to perform complex installations or upgrades, and without the need to manage and scale any hardware or storage. With fully elastic scalability, Sumo Logic is a fit for any size deployment.

What's the difference between Collector types?

Installed Collectors and Hosted Collectors are used in much different ways:

 Installed Collector	 Hosted Collector
<ul style="list-style-type: none">• <u>Installed</u> on a system within your deployment.• Can be configured with any of the following <u>Source types</u>: Local File, Remote File, Syslog, Local Windows Event Logs, Remote Windows Event Logs, and Script.	<ul style="list-style-type: none">• Hosted by Sumo Logic; no software to install or activate on a system in your deployment.• Can be configured with <u>Amazon S3 Sources</u> or <u>HTTP Sources</u> to upload data.

Installed Collectors

Installed Collectors are configured on machines in your deployment. There are a few steps in this process:

[Step 1. Download the Collector](#)

[Step 2. Install the Collector](#)

[Step 3. Add a Source](#)

[Step 4. Complete the Setup](#)

You can configure any of the following Sources on an Installed Collector:

- [Local File Source](#)
- [Remote File Source](#)
- [Local Windows Event Source](#)
- [Remote Windows Event Log Source](#)
- [Script Source](#)
- [Syslog Source](#)
- [Script Action](#)

Installed Collector Requirements

Sumo Logic Local Collectors are lightweight applications with few hardware requirements. To ensure your system can connect to the Sumo Logic service, you can check system requirements and do a quick connectivity test.

Collector memory guidelines

The minimum memory requirement for a system where you're going to install a Local Collector is 128MB, but depending on the OS running on the machine, and whether the machine is 32 bit or 64 bit, a Collector may require a significantly higher amount of memory. It's a good idea to have 256MB to 512MB available in case it's needed.

How can I check for memory issues with a Collector?

Each Collector outputs logs in the `[/install_dir]/logs/` directory. The log file that provides the most information about memory issues is named `collector.log`. You can review the log for any memory errors. If you see any Out of Memory errors in the logs you can increase the memory granted to the Collector.

System requirements

These are the minimum requirements for a system running a Sumo Logic Collector.

Hardware:

- Single core, 512MB RAM
- 256MB (minimum) available memory
- 8GB disk space

Supported Operating Systems:

- Windows, 32bit or 64bit
- Windows Server, 2008 or 2012
- Linux, major distributions, 32bit or 64bit
- MacOS 10.X
- Solaris x86
- Generic Unix capable of running Java 1.6

Supported Browsers:

- Chrome version 21 or higher
- Firefox version 14 or higher
- Safari version 5 or higher
- Internet Explorer version 9 or 10

Where should I set up Installed Collectors?

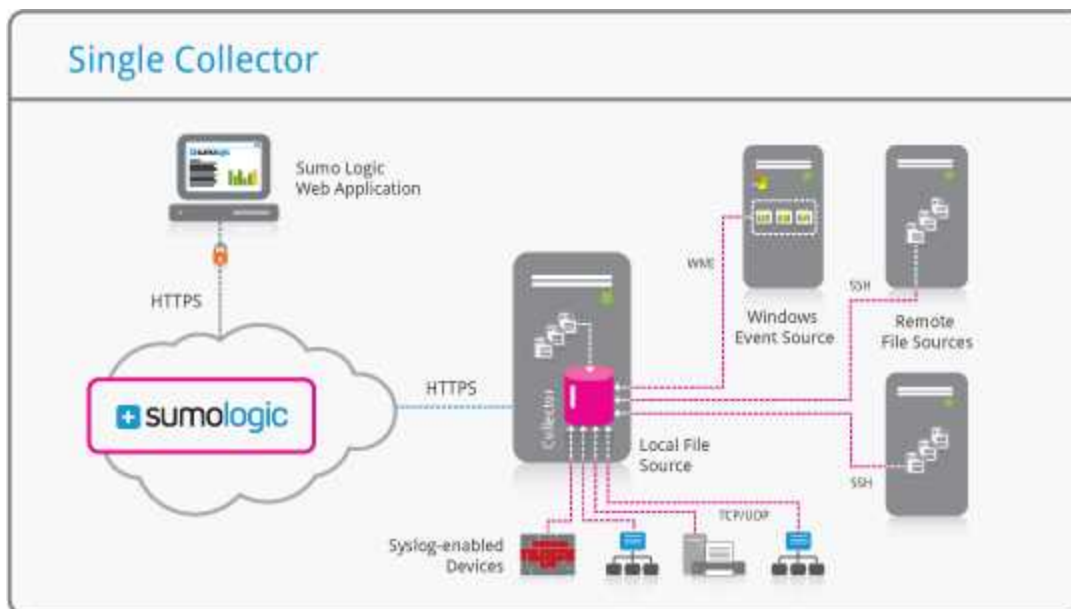
When you are deciding where to install Collectors, you have a large number of options to choose from. When you are thinking about your log collections, consider your network topology, your available bandwidth, and your domains or user groups.



Unlike Installed Collectors, Hosted Collectors don't require installation or activation. Nor do Hosted Collectors have physical requirements since Sumo Logic hosts them in AWS. For more information, see [Setting up a Hosted Collector](#).

Single Installed Collector configuration

A Sumo Logic Installed Collector can be installed on any standard server that you use for log aggregation or other network services. For example, you might decide to centralize all your collections with just one Collector installed on a dedicated machine, especially if all of your data can be accessed from a single network location. Each Installed Collector can manage about 500 files combined at a rate no greater than 15,000 events per second.



Multiple Installed Collector configuration

Have a distributed network topology? You can install multiple Collectors on many machines and set up any combination Sources to collect from your infrastructure.



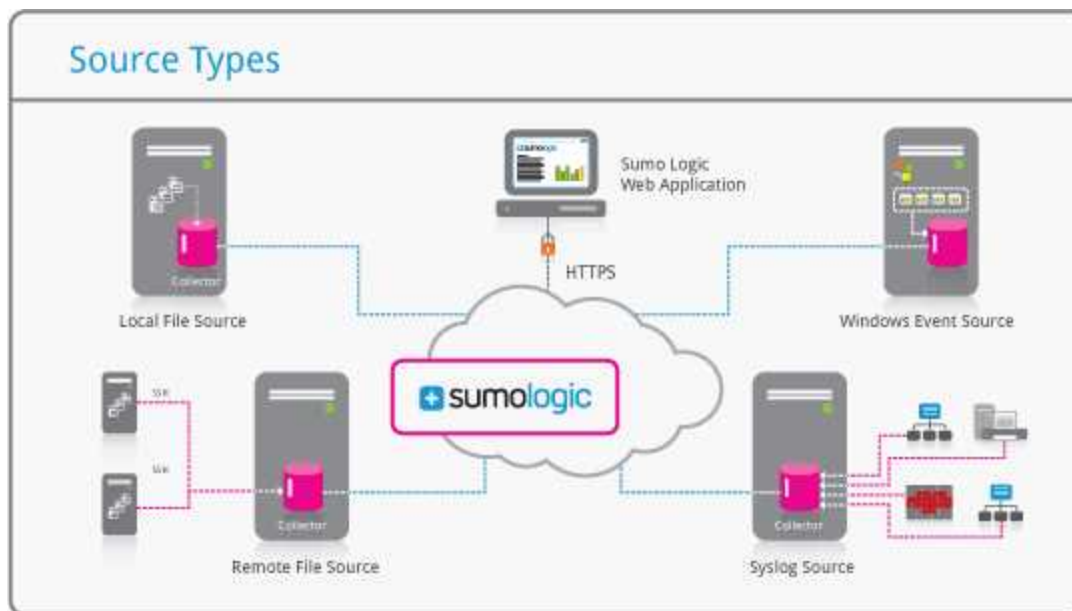
Cloud or data center deployment

Installed Collectors can also be deployed across a cloud or data center configuration—Collectors on each machine report into Sumo Logic independently sending distinct log data so that you can query against any virtual machine or server in your deployment.

Configuring Sources

After installing a Collector, the next step is to configure Sources. A Source gathers and sends logs to the Sumo Logic Cloud. You can set up as many Sources as you need for a given Collector. A Source should be configured to collect similar data types. For example, you might set up three Local File Sources to collect router activity logs from three locations, and another Local Source to collect logs from a web application.

Each Source is tagged with its own metadata. The more Sources you set up, the easier it is to isolate one of the Sources in a search since each Source can be identified by its metadata.



Collecting Local Files

For many types of collections, such as collecting output files from an application or web server, installing a Sumo Logic Collector on the same machine where the files reside is the best solution. However, for any file that can be collected locally, a remote option is always available.

Collecting Remote Files

Remote File Sources are supported using SSH (Secure Shell). For remote file collections, you will need to install the Sumo Logic Collector on a network host that has network connectivity to all the remote hosts from which you wish to collect logs. Configure the Collector with a set of valid credentials (user and password or SSH key) for those remote hosts.

Since the volume of data transferred may be large, we recommend installing the Collector on a host that is on the same network segment as the applications and devices you are collecting from.

Collecting from Syslog Sources

A Sumo Logic Syslog Source operates like a syslog server listening on a designated port to receive syslog messages. You can install the Sumo Logic Collector on the same system where your devices are currently pointing and where your current syslog server is running. Configure a Syslog Source on that Collector to listen on the same port using the same protocol as your existing syslog server. Then turn off your syslog server. All devices that were sending to your syslog server will now send data to your Sumo Logic Collector. Sumo Logic Collectors accept syslog messages over UDP or TCP.

Alternatively, you can collect files from a syslog server currently in operation. You can install a local Collector on the same machine as the existing syslog server, then set up the syslog server to output a local file. You can then collect the consolidated log file using a Local File Source.

Collecting Windows Events

Sumo Logic offers both [Local Windows Events](#) and [Remote Windows Events](#) Sources to collect the events listed in the Windows Events Viewer. Local Windows logs can be collected using a Local File Source. Logs can also be collected remotely using a CIFS/SMB path. For these collections, a Local File Source is configured to collect remote files from a UNC Share Path. Windows Events Sources can collect from the local machine, or from a remote machine within the same Domain.

Collecting from a Script

[Script Source](#) collection is the most recent addition to Sumo Logic. A Script Source isn't limited to collecting log data. If, for example, you want to poll a database, you can write a script that sends these unconventional data types to the Sumo Logic cloud.

Installed Collector volume guidelines and hardware specs

When considering the amount of data you'll be collecting with Sumo Logic it can be very helpful to understand the volume an Installed Collector can handle.

When viewing the information below, please note the following:

- Volume is measured in messages per second (MPS). This value depends on the type of Collector as well as the type of Source configured to send data to the Collector.
- A Medium Local Collector approximates the performance of a single-CPU system.
- A Large Local Collector approximates the performance of a dual-CPU system.
- An XLarge Local Collector approximates the performance of a quad-CPU system.
- In all cases, CPU data below is listed in absolute terms, meaning that a Medium Collector has a max CPU usage of 100%; a Large Collector has a max CPU usage of 200%, and an XLarge Collector has a max CPU usage of 400%.

Linux (64 bit) Collectors

Local File Source limits

Collector size	Message Per Second limit	Max Heap	CPU usage
Medium	8,750	128MB	80%
Large	10,000	128MB	105%
XLarge	10,000	128MB	115%

Syslog Source limits (TCP)

Collector size	Message Per Second limit	Max Heap	CPU usage
Medium	6,000	128MB	80%
Large	15,000	128MB	160%
XLarge	20,000	128MB	290%

Syslog Source limits (UDP)

Collector size	Message Per Second limit	Max Heap	CPU usage
Medium	5,000	128MB	70%
Large	12,500	128MB	130%
XLarge	17,500	128MB	190%

Remote File Source limits

Collector size	Message Per Second limit	Max Heap	CPU usage
Medium	3,100	128MB	45%

Windows 2008 (64 bit) Collectors

Local File Source limits

Collector size	Message Per Second limit	Max Heap	CPU usage
Medium	4,250	128MB	60%

Event Log Source limits

Collector size	Message Per Second limit	Max Heap	CPU usage
Medium	250	128MB	45%

Testing connectivity

To ensure you have the required access and connectivity to the Sumo Logic service, there are a couple of checks you can do from the machine where you want to install a Collector.

To check connectivity, try one of these two options:

- Open a browser to <https://collectors.sumologic.com>. You should see the word "Tweep".



- Try connecting via telnet. Telnet to **collectors.sumologic.com 443**. You should see a result similar to:

```
$telnet collectors.sumologic.com 443
Trying collectors.sumologic.com...
Connected to prod-events-lb-1056629993.us-east-1.elb.amazonaws.com.
```

If results show a failure to connect, check your firewall settings.

If Windows Firewall is enabled, create an Outbound Rule to allow outbound Collector connections and then try again. To create a new Outbound Rule for your Windows firewall:



These are instructions for a Windows 2008r2 system. If you are installing on another version of Windows, the steps should be similar, but are not likely to be exact.

1. Go to **Administrative Tools > Firewall Settings**.
2. Click **New Rule**.
3. Select **Port** for the type of rule, and then click **Next**.
4. Enter **443** for the port, and then click **Next**.
5. Select **Allow** for the connection and then click **Next**.
6. Select the appropriate Profile for your environment. Leave all three options checked (recommended): Domain, Public, and Private, and then click **Next**.
7. Enter a name and a description for your new rule, and then click **Finish**.

Downloading Installed Collector Software

In the following steps we'll download the Collector from the Sumo Logic Web Application.



You can also download the Collector software from a static URL.

To download the Collector:

1. In the Sumo Logic Web Application select **Manage > Collectors**.
2. Click **Add Collector**.



3. Click **Installed Collector**.



4. Click the version of the installer you need:



The installer begins to download immediately. Then it's time to install the Collector.

Downloading a Collector from a static URL

Use the following URLs to download the most recent version of a Collector. The URLs are static; the version of the Collector will be updated each time there is a release or patch.

To download a Collector directly from a URL:

Choose one of the following:

- **Linux 32:** <https://collectors.sumologic.com/rest/download/linux/32>
- **Linux 64:** <https://collectors.sumologic.com/rest/download/linux/64>
- **Linux Debian:** <https://collectors.sumologic.com/rest/download/deb/64>
- **Linux RPM:** <https://collectors.sumologic.com/rest/download/rpm/64>
- **Mac OS:** <https://collectors.sumologic.com/rest/download/macos>
- **Solaris:** <https://collectors.sumologic.com/rest/download/solaris/32>
- **Tarball:** <https://collectors.sumologic.com/rest/download/tar>
- **Windows 32:** <https://collectors.sumologic.com/rest/download/windows>
- **Windows 64:** <https://collectors.sumologic.com/rest/download/win64>

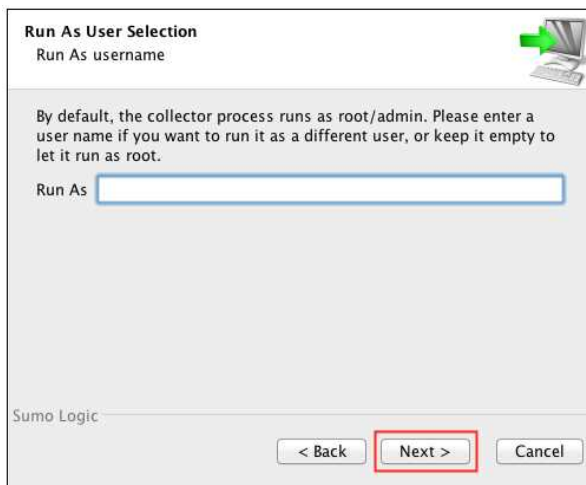
Installing a Collector

Make sure to install the Collector on a server with network connectivity to the machines from which you want to gather files. You'll want to run the installer on your server with root or Administrator privileges.

To install the Collector:

1. Double-click the installer.
2. Accept the terms of use and click **Next**.
3. Accept the default installation location, or specify a different location. Click **Next**.
4. When asked if you'd like to **Run As Another User**, do one of the following:
 - If you want to install the Collector so that it runs as root of your server, just click **Next**.
 - If you want to install the Collector using a specific user account, type the name of the account and click **Next** (Mac/Linux). On Unix, type the user account name. On Windows, you'll need to type the name and password of the user account.

Note: Choosing to run as a user means that the Collector won't be able to gather logs that are only readable by a root.



5. If you'd like to choose a default Source at this point, choose the file and click **Next**. Otherwise, you can set up Sources at a later time.



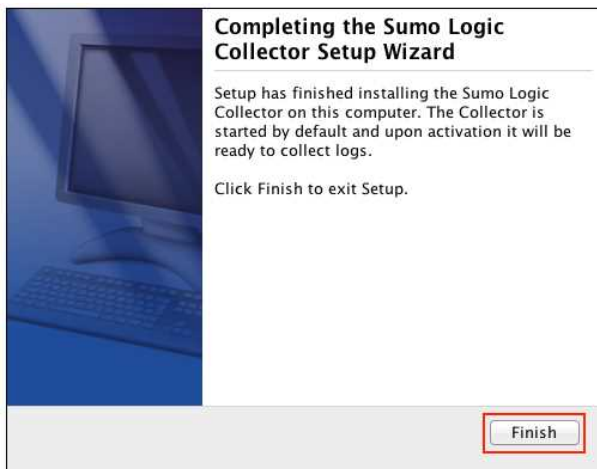
6. To register the Collector, choose the way the it will be authenticated before clicking **Next**:
 - Credentials:** To use email and password credentials, select Credentials, then type the information in the text boxes.
 - Access Key:** To use a generated Access Key and Access ID combination to authenticate, select Access

Key and then copy and paste the information in the text boxes.



The image shows a dialog box titled "Sumo Logic Account Credential" with a sub-header "Account Credential". It has two radio buttons: "Credentials" (unselected) and "Access Key" (selected). Under "Credentials", it says "Please enter your Sumo Logic Account email and password." with fields for "Email" and "Password". Under "Access Key", it says "Please enter your Sumo Logic Access Id and Access Key." with fields for "Access Id" (containing a long alphanumeric string) and "Access Key" (containing dots). At the bottom, there is a "Sumo Logic" logo and "Next >" and "Cancel" buttons.

7. When the setup app has completed, click **Finish**. The Collector is added to your account (check the Collectors page to confirm).

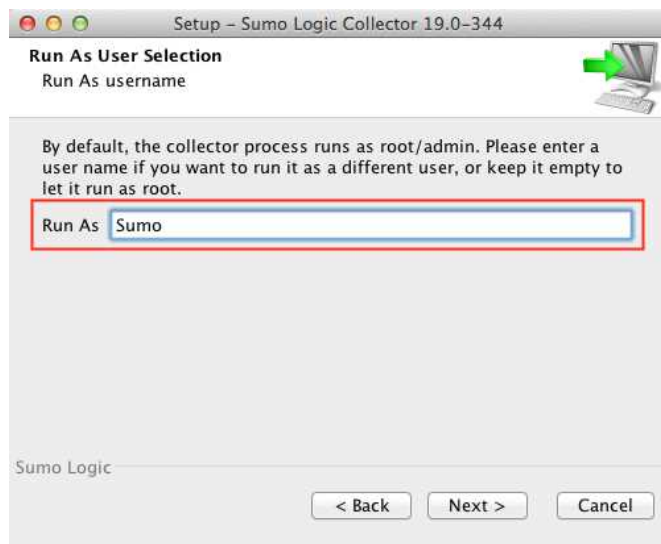


You're now ready to add a Source to your Collector.

Running a Collector on a User Account

During installation, you can choose to run as the root or you can choose to run the Collector from a specific user account. This option can be very helpful in situations where you don't want offer Sumo Logic full access to the computer where the Collector is installed; instead the Collector assumes the limitations and privileges of the user account.

It may make sense to create a new user account that has privileges only to the directories that you'd like the Collector to access. For example, you could create an account named Sumo, and assign permissions that work with your security policies but still allow access to appropriate data:



Windows only: You'll need to provide both the user account name and a valid password for that account:



Please note the following:

- During set up, you must type a valid user name in the dialog box. If the user name is incorrect, you'll need to start the install process over.
- (Windows only) The user account you choose to run must have the right to **Log on as a Service**. This can be set in Administrative Tools > Local security Policy > User Rights Assignment; add the user for Log on as a Service.
- (Windows only) When it's time to upgrade the Collector, you'll need to enter a valid password for the user account. If you've forgotten the password, make sure to reset the password before upgrading.

Tarball installation for Linux

Sumo Logic offers a generic tarball file that can be used to script unattended Collector installations for one or more Collectors. You'll need to download the tarball on each machine where you'll install a Collector and make sure that JSON has been configured for the Sources you'll use to collect data on each machine.

In order to install the Collector, you'll need to make sure that you've created an appropriate user account on each computer, with admin permissions in addition to Sumo Logic account credentials. These credentials are used during installation to [register the Collector](#).

Step 1. Download the tarball

The tarball file includes all the required files for installing and activating a Sumo Logic Collector *except* the Java Runtime Environment (JRE) Version 7.

To download the tarball:

Do one of the following:

- Download tar.gz from the Sumo Logic Web Application (Download Collector > Generic Unix).
- If you're using an installation script, use wget or curl to download the file from:
<https://collectors.sumologic.com/rest/download/tar>.



If you're installing on an AMI, create an image that includes the tarball.

Preparing the tarball:

1. Do `tar -xvf tar` to create the /sumocollector directory:
2. Move /sumocollector/tanuki/Linux/wrapper to /sumocollector/wrapper.



If the wrapper isn't in the correct directory, the "Unable to locate any of the following binaries" message displays.

4. Move the other file to {version}/bin/native/lib.
5. Run the following to allow the wrapper and collector executables to run:

```
chmod ug+x wrapper
```

```
chmod ug+x collector
```

Step 2. Configure Sources

The Collector Management API allows you to pass all Source settings in a JSON file, including naming a Source, adding metadata tags, and pointing the Source to the files you want to collect. For instructions, see [Using JSON to configure Sources](#).

Optionally, you can configure Sources using the Sumo Logic Web Application after the Collector has been installed.

Step 3. Modify wrapper.conf

Now you'll modify the wrapper.conf file to point to the location where the JRE exists on the computer where you'll install the Collector.

To modify wrapper.conf:

1. In the installation directory, find the wrapper.conf file here: `[installation dir]/config/wrapper.conf`.
2. Open the `wrapper.conf` file and locate the `wrapper.java.command` line that looks like this:

```
# Java Application
# Locate the java binary on the system PATH:
#wrapper.java.command=java
# Specify a specific java binary:
wrapper.java.command=${JAVA_COMMAND_LOCATION}
```

3. Modify this line to point to the Java command location for the target operating system. For example:

```
# Java Application
# Locate the java binary on the system PATH:
#wrapper.java.command=java
# Specify a specific java binary:
wrapper.java.command=c:\Program Files (x86)\Java\jre6\bin
```

```
# Java Application
# Locate the java binary on the system PATH:
#wrapper.java.command=java
# Specify a specific java binary:
wrapper.java.command=/opt/java-1.6.0_amd/bin/java
```

Step 4. Create sumo.conf

The `sumo.conf` file allows you to pass Collector configuration parameters to Sumo Logic during installation. Once installation is complete, `sumo.conf` isn't needed (compared to `wrapper.conf`, which cannot be deleted).

There are two ways you can pass credentials that will be used to authenticate Collectors. You can either include email and password parameters, or you can include `accessid` and `accesskey` parameters to use a Access ID and Access Key credentials that are generated through the Sumo Logic Web Application.

Only one set of credentials are needed (either email/password or accessid/accesskey).

Required parameters in `sumo.conf` are:

- **name**. Name of the Collector. If left blank, the host name will be used.
- **sources**. Path to JSON file that contains Source configuration. (See [Using JSON to configure Sources](#).)
- **accessid**. Access ID credential used to register the Collector.
- **accesskey**. Access Key credential.
- **email**. Email address used to register the Collector.
- **password**. Password used to register the Collector.

Remember, if you use accessid and accesskey, you should not include email and password. Conversely, if you use email and password, do not include accessid and accesskey.

To create **sumo.conf**:

1. Using a text editor (or any similar program) create a new file.
2. Add all of the required parameters (for more information see [sumo.conf Parameters](#)).
3. Save the file to **/etc/sumo.conf**.

Step 5. Run the Collector

The Collector is activated the first time you execute it.

Control the starting and stopping of the service by using:

***nix:** `sudo /usr/local/SumoCollector/collector]`
{start|stop|status|restart}

Installing a Collector on Linux with RPM

The RPM installer uses a collector script to install a Collector to run as root on Linux 64 systems. By default, RPM installs the Collector in the `/opt/SumoCollector` directory.



If you're running a Linux 32 system, don't use the RPM installer. Follow [these instructions](#) instead.

About updating your configuration

If you upgrade a Collector using RPM, no configuration changes made to `sumo.conf` will be passed to the Collector. Instead, you can modify the "collector" script (found in the RPM package) to pass any changes you'd like to make.

Step 1. Download the RPM package

The RPM package is downloaded from a static URL, found here:

<https://collectors.sumologic.com/rest/download/rpm/64>

Step 2. Configure Sources

The Collector Management API allows you to pass all Source settings in a JSON file, including naming a Source, adding metadata tags, and pointing the Source to the files you want to collect. For instructions, see [Using JSON to configure Sources](#).

Alternately, you can create Sources after the Collector has been installed using the Sumo Logic Web Application.

Step 3. Configure `sumo.conf`

The `sumo.conf` file allows you to pass Collector configuration parameters to Sumo Logic during installation. Once installation is complete, `sumo.conf` isn't needed (compared to `wrapper.conf`, which cannot be deleted).

There are two ways you can pass credentials that will be used to authenticate Collectors. You can either include email and password parameters, or you can include `accessid` and `accesskey` parameters to use a [Access ID and Access Key credentials](#) that are generated through the Sumo Logic Web Application. **Only one set of credentials are needed (either email/password or accessid/accesskey).**

Required parameters in `sumo.conf` are:

- **name.** Name of the Collector. If left blank, the host name will be used.
- **sources.** Path to JSON file that contains Source configuration. (See [Using JSON to configure Sources](#).)
- **accessid.** Access ID credential used to register the Collector.
- **accesskey.** Access Key credential.
- **email.** Email address used to register the Collector.
- **password.** Password used to register the Collector.

To create `sumo.conf`:

1. Using a text editor (or any similar program) create a new file.
2. Add all of the required parameters (for more information see [sumo.conf Parameters](#)).
3. Save the file to `/etc/sumo.conf`.

To change from root to another user:

- In `sumo.conf`, add the following:
`rpmRunAs=newusername` (replacing `newusername` with the user account you'd like Sumo Logic to install under)

Make sure to not add any extra spaces; entries are case sensitive.

To disable the starting of the Collector process as a part of installation:

- In `sumo.conf`, add the following
`rpmAutoStart=false`

Step 4. Install the Collector

If you have not changed the default auto-start of the Collector process, step 2 is not required, because the Collector will automatically be started.

1. Launch the RPM package to install the Collector.

By default, RPM installs the Collector in the `/opt/SumoCollector` directory.

2. (Optional) Start the Collector by running: `/etc/init.d/collector start`. This step is only needed if you've disabled the automatic starting of the Collector in `sumo.conf`.

Unattended installation from a Linux script using the Collector Management API

There are three methods of silent (unattended) installation for Linux computers:

Scripted installation. The instructions for command line/scripted installation are included below.

RPM. The Sumo Logic RPM installer is the quickest method, but you may find you'll need to modify some settings after the install is complete. For more information and instructions, see [Installing a Collector with RPM](#)

Tarball. The generic tarball installer doesn't include the Java Runtime Environment; if you're using a non-standard flavor of the JRE on your Linux box, the tarball installation method is a great option. For more information and instructions, see [Silent installation](#).

Step 1. Download the Collector

Before installing, you will need to choose whether to download the 32-bit version or the 64-bit version of the installer file. To determine whether you need 32 or 64 bit, open a CLI on the target system and type: `uname -a`. If the result mentions "x86_64" or "amd64" download the 64-bit installer; otherwise download the 32-bit installer.

To download the Collector:

Do one of the following:

- Download the **32-bit** Linux Collector directly from:
<https://collectors.sumologic.com/rest/download/linux/32>.
- Download the **64-bit** Linux Collector directly from:
<https://collectors.sumologic.com/rest/download/linux/64>.
- From the Sumo Logic Web Application, go to the Collectors tab, and click **Add Collector**. Then click **Installed Collector** and download to a desktop, then copy the binary file to the target system using `scp` or `SFTP`.

Step 2. Configure `sumo.conf`

The first time you install a Collector, the `sumo.conf` file allows you to pass Collector configuration parameters to Sumo Logic during installation. Once installation is complete, `sumo.conf` isn't needed (compared to `wrapper.conf`, which cannot be deleted).

There are two ways you can pass credentials that will be used to authenticate Collectors. You can either include email and password parameters, or you can include `accessid` and `accesskey` parameters to use a [Access ID and](#)

Access Key credentials that are generated through the Sumo Logic Web Application. **Only one set of credentials are needed (either email/password or accessid/accesskey).**

Required parameters in sumo.conf are:

- **name.** Name used to register the Collector. If left blank, the host name will be used.
- **email.** Email used to register the Collector.
- **accesskey.** Access Key credential.
- **accessid.** Access ID credential used to register the Collector.
- **password.** Password used to register the Collector.
- **sources.** Path to JSON file that contains Source configuration.

To create sumo.conf:

1. Using a text editor (or any similar program) create a new file.
2. Add all of the required parameters (for more information see [sumo.conf Parameters](#)).
3. Save the file to `/etc/sumo.conf`.

Step 3. Configure Sources using the Collector Management API

Instead of using installation directory's JSON settings, you can pass Source parameters in an array using the Collector Management API.

To see a list of parameters for each Source type, see the [Collector Management API documentation](#)

Step 4. Install the Collector

Before installing the Collector, make sure root has executable privileges for the file by typing: `chmod 740 SumoCollector_[os-type]_[build-date]_xxxxx.sh`

To install the Collector:

- From the download directory, as the root user, run the installation file `SumoCollector_[os-type]_[build-date]_xxxxx.sh -q`

The Collector runs as a service and starts automatically after installing or rebooting. The first time the Collector starts, the credentials set in sumo.conf are used to activate and register the Collector and Sources.

Unattended installation from a Linux script

There are three methods of silent (unattended) installation for Linux computers:

Scripted installation. The instructions for command line/scripted installation are included below.

RPM. The Sumo Logic RPM installer is the quickest method, but you may find you'll need to modify some settings after the install is complete. For more information and instructions, see [Installing a Collector with RPM](#)

Tarball. The generic tarball installer doesn't include the Java Runtime Environment; if you're using a non-standard flavor of the JRE on your Linux box, the tarball installation method is a great option. For more information and instructions, see [Silent installation](#).

Step 1. Download the Collector

Before installing, you will need to choose whether to download the 32-bit version or the 64-bit version of the installer file. To determine whether you need 32 or 64 bit, open a CLI on the target system and type: `uname -a`. If the result mentions "x86_64" or "amd64" download the 64-bit installer; otherwise download the 32-bit installer.

To download the Collector:

Do one of the following:

- Download the **32-bit** Linux Collector directly from:
<https://collectors.sumologic.com/rest/download/linux/32>.
- Download the **64-bit** Linux Collector directly from:
<https://collectors.sumologic.com/rest/download/linux/64>.
- From the Sumo Logic Web Application, click the **Download Collector** link from the **Collectors** tab; download to a desktop, then copy the binary file to the target system using `scp` or `SFTP`.

Step 2. Configure Sources

The Collector Management API allows you to pass all Source settings in a JSON file, including naming a Source, adding metadata tags, and pointing the Source to the files you want to collect. For instructions, see [Using JSON to configure Sources](#).

Optionally, you can configure Sources using the Sumo Logic Web Application after the Collector has been installed.

Step 3. Configure sumo.conf

The `sumo.conf` file allows you to pass Collector configuration parameters to Sumo Logic during installation. Once installation is complete, `sumo.conf` isn't needed (compared to `wrapper.conf`, which cannot be deleted).

There are two ways you can pass credentials that will be used to authenticate Collectors. You can either include email and password parameters, or you can include `accessid` and `accesskey` parameters to use a [Access ID and Access Key](#) credentials that are generated through the Sumo Logic Web Application. **Only one set of credentials are needed (either email/password or accessid/accesskey).**

Required parameters in `sumo.conf` are:

- **name**. Name of the Collector. If left blank, the host name will be used.
- **sources**. Path to JSON file that contains Source configuration. (See [Using JSON to configure Sources](#).)
- **accessid**. Access ID credential used to register the Collector.
- **accesskey**. Access Key credential.
- **email**. Email address used to register the Collector.
- **password**. Password used to register the Collector.

To create `sumo.conf`:

1. Using a text editor (or any similar program) create a new file.
2. Add all of the required parameters (for more information see [sumo.conf Parameters](#)).

3. Save the file to `/etc/sumo.conf`.

Step 4. Install the Collector

Before installing the Collector, make sure root has executable privileges for the file by typing: `chmod 740 SumoCollector_[os-type]_[build-date]_xxxxx.sh`. You can run the install command with `-q` to install the Collector with default settings and location.

To install the Collector:

- From the download directory, as the root user, run the installation file `SumoCollector_[os-type]_[build-date]_xxxxx.sh -q`

The Collector runs as a service and starts automatically after installing or rebooting. The first time the Collector starts, the credentials set in `sumo.conf` are used to activate and register the Collector and Sources.

Silent Installation for Mac and Unix using the Tarball

Sumo Logic offers a generic tarball file that can be used to script unattended Collector installations for one or more Collectors. You'll need to download the tarball on each machine where you'll install a Collector and make sure that JSON has been configured for the Sources you'll use to collect data on each machine.

In order to install the Collector, you'll need to make sure that you've created an appropriate user account on each computer, with admin permissions in addition to Sumo Logic account credentials. These credentials are used during installation to register the Collector.

Step 1. Download the tarball

The tarball file includes all the required files for installing and activating a Sumo Logic Collector *except* the Java Runtime Environment (JRE) Version 7.

To download the tarball:

Do one of the following:

- Download `tar.gz` from the Sumo Logic Web Application (Download Collector > Generic Unix).
- If you're using an installation script, use `wget` or `curl` to download the file from:
<https://collectors.sumologic.com/rest/download/tar>.



If you're installing on an AMI, create an image that includes the tarball.

Preparing the tarball:

1. Do `tar -xvf tar` to create the `/sumocollector` directory:
2. Move `/sumocollector/tanuki/{OS Type}/wrapper` to `/sumocollector/wrapper`.



If the wrapper isn't in the correct directory, the "Unable to locate any of the following binaries" message displays.

4. Move the other file to {version}/bin/native/lib.
5. (Mac only) Run the following to allow the wrapper and collector executables to run:

```
chmod ug+x wrapper
```

```
chmod ug+x collector
```

Step 2. Configure Sources

The Collector Management API allows you to pass all Source settings in a JSON file, including naming a Source, adding metadata tags, and pointing the Source to the files you want to collect. For instructions, see [Using JSON to configure Sources](#).

Optionally, you can configure Sources using the Sumo Logic Web Application after the Collector has been installed.

Step 3. Modify wrapper.conf

Now you'll modify the wrapper.conf file to point to the location where the JRE exists on the computer where you'll install the Collector.

To modify wrapper.conf:

1. In the installation directory, find the wrapper.conf file here: [installation dir]/config/wrapper.conf.
2. Open the wrapper.conf file and locate the wrapper.java.command line that looks like this:

```
# Java Application
# Locate the java binary on the system PATH:
#wrapper.java.command=java
# Specify a specific java binary:
wrapper.java.command=${JAVA_COMMAND_LOCATION}
```

3. Modify this line to point to the Java command location for the target operating system. For example:

```
# Java Application
# Locate the java binary on the system PATH:
#wrapper.java.command=java
# Specify a specific java binary:
wrapper.java.command=c:\Program Files (x86)\Java\jre6\bin
```

```
# Java Application
# Locate the java binary on the system PATH:
#wrapper.java.command=java
# Specify a specific java binary:
wrapper.java.command=/opt/java-1.6.0_amd/bin/java
```

Step 4. Create sumo.conf

The **sumo.conf** file allows you to pass Collector configuration parameters to Sumo Logic during installation. Once installation is complete, sumo.conf isn't needed (compared to wrapper.conf, which cannot be deleted).

There are two ways you can pass credentials that will be used to authenticate Collectors. You can either include email and password parameters, or you can include accessid and accesskey parameters to use a [Access ID and Access Key credentials](#) that are generated through the Sumo Logic Web Application.

Only one set of credentials are needed (either email/password or accessid/accesskey).

Required parameters in sumo.conf are:

- **name.** Name of the Collector. If left blank, the host name will be used.
- **sources.** Path to JSON file that contains Source configuration. (See [Using JSON to configure Sources.](#))
- **accessid.** Access ID credential used to register the Collector.
- **accesskey.** Access Key credential.
- **email.** Email address used to register the Collector.
- **password.** Password used to register the Collector.

Remember, if you use accessid and accesskey, you should not include email and password. Conversely, if you use email and password, do not include accessid and accesskey.

To create sumo.conf:

1. Using a text editor (or any similar program) create a new file.
2. Add all of the required parameters (for more information see [sumo.conf Parameters](#)).
3. Save the file to **/etc/sumo.conf**.

Step 5. Run the Collector

The Collector is activated the first time you execute it.

To execute the Collector:

Control the starting and stopping of the service by using:

***nix: `sudo /usr/local/SumoCollector/collector]`**

`{start|stop|status|restart}`

Unattended installation for Windows

With all silent (unattended) installations, you'll first download the Collector, then configure a Source using a JSON file stored in the installation directory.

Step 1. Download the Collector

Do one of the following:

- In the Sumo Logic Web Application, click the Collectors tab, then click the **Download Collector** to download the **Windows** installer.
- Download the Collector directly from <https://collectors.sumologic.com/rest/download/windows>.

Step 2. Configure Sources

The Collector Management API allows you to pass all Source settings in a JSON file, including naming a Source, adding metadata tags, and pointing the Source to the files you want to collect. For instructions, see [Using JSON to configure Sources](#).

Optionally, you can configure Sources using the Sumo Logic Web Application after the Collector has been installed.

Step 3. Configure sumo.conf

The [sumo.conf](#) file allows you to pass Collector configuration parameters to Sumo Logic during installation. Once installation is complete, sumo.conf isn't needed (compared to wrapper.conf, which cannot be deleted).

There are two ways you can pass credentials that will be used to authenticate Collectors. You can either include email and password parameters, or you can include accessid and accesskey parameters to use a [Access ID and Access Key credentials](#) that are generated through the Sumo Logic Web Application. **Only one set of credentials are needed (either email/password or accessid/accesskey).**

Required parameters in sumo.conf are:

- **name.** Name of the Collector. If left blank, the host name will be used.
- **sources.** Path to JSON file that contains Source configuration. (See [Using JSON to configure Sources](#).)

Credentials: One set of credentials is needed, either accessid/accesskey OR email/password.

- **accessid** and **accesskey**. Access ID credential used to register the Collector and Access Key credential.
- **email** and **password**. Email address and password used to register the Collector.

To create sumo.conf:

1. Using a text editor (or any similar program) create a new file.
2. Add all of the required parameters (for more information see [sumo.conf Parameters](#)).
3. Save the file to **C:\sumo\sumo.conf**.

Step 4. Install the Collector

To install the Collector:

- To install the Collector with the default settings and location, run the command with **-q**. For example:
SumoCollector_windows_13_1-4.exe -q

(Optional) To install the Collector in a specific directory:

- Use **-dir** to specify the directory. For example: **SumoCollector_windows_13_1-4.exe -q -dir "C:\Program Files\Folder\SubFolder"**

The Collector runs as a service and starts automatically after installing or rebooting. The first time the Collector starts, the credentials set in sumo.conf are used to activate and register the Collector and Sources.

Tarball installation for Windows

Sumo Logic offers a generic tarball file that can be used to script unattended Collector installations for one or more Collectors. You'll need to download the tarball on each machine where you'll install a Collector and make sure that JSON has been configured for the Sources you'll use to collect data on each machine.

In order to install the Collector, you'll need to make sure that you've created an appropriate user account on each computer, with admin permissions in addition to Sumo Logic account credentials. These credentials are used during installation to register the Collector.

Step 1. Download the tarball

The tarball file includes all the required files for installing and activating a Sumo Logic Collector *except* the Java Runtime Environment (JRE) Version 7.

To download the tarball:

Do one of the following:

- Download tar.gz from the Sumo Logic Web Application (Download Collector > Generic Unix).
- If you're using an installation script, use wget or curl to download the file from:
<https://collectors.sumologic.com/rest/download/tar>.



If you're installing on an AMI, create an image that includes the tarball.

Preparing the tarball:

1. Use 7-zip or Winzip to extract the tar file to create the `/sumocollector` directory.
 2. Move `/sumocollector/tanuki/{OS Type}/wrapper` to `/sumocollector/wrapper`. (For Windows, the file name is `wrapper.exe`.)
-



If the wrapper isn't in the correct directory, the "Unable to locate any of the following binaries" message displays.

4. Move the other file to `{version}/bin/native/lib`.

Step 2. Configure Sources

The Collector Management API allows you to pass all Source settings in a JSON file, including naming a Source, adding metadata tags, and pointing the Source to the files you want to collect. For instructions, see [Using JSON to configure Sources](#).

Optionally, you can configure Sources using the Sumo Logic Web Application after the Collector has been installed.

Step 3. Modify `wrapper.conf`

Now you'll modify the `wrapper.conf` file to point to the location where the JRE exists on the computer where you'll install the Collector.

To modify `wrapper.conf`:

1. In the installation directory, find the `wrapper.conf` file here: `[installation dir]/config/wrapper.conf`.
2. Open the `wrapper.conf` file and locate the `wrapper.java.command` line that looks like this:

```
# Java Application
# Locate the java binary on the system PATH:
#wrapper.java.command=java
# Specify a specific java binary:
wrapper.java.command=${JAVA_COMMAND_LOCATION}
```

3. Modify this line to point to the Java command location for the target operating system. For example:

```
# Java Application
# Locate the java binary on the system PATH:
#wrapper.java.command=java
# Specify a specific java binary:
wrapper.java.command=c:\Program Files (x86)\Java\jre6\bin
```

```
# Java Application
# Locate the java binary on the system PATH:
#wrapper.java.command=java
# Specify a specific java binary:
wrapper.java.command=/opt/java-1.6.0_amd/bin/java
```

Step 4. Create sumo.conf

The **sumo.conf** file allows you to pass Collector configuration parameters to Sumo Logic during installation. Once installation is complete, sumo.conf isn't needed (compared to wrapper.conf, which cannot be deleted).

There are two ways you can pass credentials that will be used to authenticate Collectors. You can either include email and password parameters, or you can include accessid and accesskey parameters to use a [Access ID and Access Key credentials](#) that are generated through the Sumo Logic Web Application.

Only one set of credentials are needed (either email/password or accessid/accesskey).

Required parameters in sumo.conf are:

- **name.** Name of the Collector. If left blank, the host name will be used.
- **sources.** Path to JSON file that contains Source configuration. (See [Using JSON to configure Sources](#).)
- **accessid.** Access ID credential used to register the Collector.
- **accesskey.** Access Key credential.
- **email.** Email address used to register the Collector.
- **password.** Password used to register the Collector.

Remember, if you use accessid and accesskey, you should not include email and password. Conversely, if you use email and password, do not include accessid and accesskey.

To create sumo.conf:

1. Using a text editor (or any similar program) create a new file.
2. Add all of the required parameters (for more information see [sumo.conf Parameters](#)).
3. Save the file to **/etc/sumo.conf**.

Step 5. Run the Collector

The Collector is activated the first time you execute it.

To execute the Collector:

1. Register and start the service by running **InstallCollector-NT.bat** to register the Collector as a service on the Windows system.
2. Run **startCollectorService.bat** to start the Collector.

Advanced Collector installation

In addition to installing via the Web Application, there are several options for silent/scripted installation.

All methods of silent or command line installation follow the same basic steps:

1. Download the Collector (either from the Web Application or a static URL).
2. Use a JSON file to pass Source configuration information to Sumo Logic.
3. Create the sumo.conf file to pass Collector credentials and parameters to Sumo Logic.
4. Complete the installation and, if necessary, restart the Collector.

A note about JRE

For Linux, Mac, Solaris, and Windows downloads, the Java Runtime Environment (JRE 7) is bundled with the installer. If you are using the Generic Unix tarball file, you must install the required JRE 7 on your target systems.

Using sumo.conf

The **sumo.conf** file allows you to pass Collector configuration parameters to Sumo Logic during installation. Once installation is complete, you may delete sumo.conf.

There are two ways you can pass credentials that will be used to authenticate Collectors. You can either include email and password parameters, or you can include accessid and accesskey parameters to use a Access ID and Access Key credentials that are generated through the Sumo Logic Web Application. **Only one set of credentials are needed (either email/password or accessid/accesskey).**

A typical sumo.conf file looks something like this:



sumo.conf parameters

The following parameters can be passed to Sumo Logic using sumo.conf. If any parameters are missing, Sumo Logic looks for the parameters in wrapper.conf. A recommended best practice is to always pass email and

password credentials through sumo.conf so that you can delete the file later.

Parameter	Description	Example
name	Name used to register the Collector. If no name is specified, the host name is used.	name=FirewallLogs
email	Email used to register the Collector.	email=name@company.com
password	Password used to register the Collector.	password=1234abcD
accessid	Access ID generated in the Security page. (See Generating Access API Keys.)	accessid=MboxeezMzN8S
accesskey	Access Key generated in the Security page. (See Generating Access API Keys.)	accesskey=dBorwTn8TxF8ofounEXnsQ4hPpuqCzw
sources	Path to JSON file that contains Source configuration. (See JSON Source Configuration.) Important: On Windows the path value for "sources=" must be specified with double slashes, \\, as shown in the example.	sources=c:\\sumo\\sources.json
override	Overrides a Collector's existing Sources to delete all Sources except for the one specified by the sources parameter (above). Don't include in sumo.conf unless you specifically need to use this flag.	override=true
ephemeral	Sets the Collector as ephemeral. Don't include in sumo.conf unless you'd like to flag the Collector as ephemeral. (See Automatically deleting offline Collectors with ephemeral.)	ephemeral=true
clobber	Sets the clobber flag; don't include unless you'd like to clobber the Collector. (See Forcing or overwriting a Collector's name with clobber.)	clobber=true

Creating sumo.conf

After downloading the Collector installer, you'll create the sumo.conf file in a specific directory. When installation begins, Sumo Logic first looks for sumo.conf; if sumo.conf isn't found, Sumo Logic then looks to wrapper.conf for these parameters. Using sumo.conf is preferable because it can be deleted, which protects your credentials; wrapper.conf cannot be deleted, so any parameters you enter must be retained (including the email and password used to register the Collector).

To create sumo.conf:

1. Using a text editor (or any similar program) create a new file.
2. Add all of the required parameters, and any optional parameters, listed in [sumo.conf Parameters.](#)
3. Save the file to `/etc/sumo.conf` (Mac or Linux) or `C:\\sumo\\sumo.conf` (Windows).

Using sumo.conf to disable Collector options

In addition to configuring Sources, sumo.conf can be used to pass parameters to disable certain Collector options.



Because these options are disabled by adding a parameter to **sumo.conf** this attribute must be configured at the time a Collector is installed.

Disabling remote Collector upgrades

If your policies restrict software upgrades to a specific schedule, an Admin can disable any remote upgrades to Collectors. This option allows your organization to enforce any policies by removing the **Upgrade Collector** link from the Sumo Logic Web Application. Instead, an Admin needs to upgrade the Collector manually using one of the [Advanced Installation](#) methods.

To disable remote Collector upgrades:

- Add `disableUpgrade=true` to **sumo.conf**.

Disabling Source types

If your organization's internal policies restrict the use of scripts, you can use sumo.conf to disable the creation of script-based Script Sources and Script Action Sources. When this parameter is passed, those two options are removed from the Sumo Logic Web Application; additionally, those Source types cannot be configured via sumo.conf.

To disable remote Collector upgrades:

Add one or both of the following to sumo.conf:

- `disableScriptSource=true`
- `disableActionSource=true`

Forcing a Collector's Name with Clobber

During a scripted or manual installation, you can use the clobber flag in situations where you're creating a new Collector that will use a name that is already in use by another Collector. Clobber is only effective before the new Collector has been registered (activated) with Sumo Logic.



Using the clobber flag deletes any existing **Collector** with the same name, so proceed with extreme caution.

You can choose to either modify the `wrapper.conf` file, or create a new `sumo.conf` file to pass the clobber flag to Sumo Logic.

If you clobber a Collector, then reuse the Collector's name, note that you can't search logs from the clobbered Collector using the `_collector=<name>` or `_source=<name>` keywords, but you can search the old logs using `_sourceName`, `_sourceCategory`, `_sourceHost`, or by generic search by time. Even though the name matches the clobbered Collector, each Collector (and Source) are tagged with a unique ID that isn't reused.

To set the clobber flag in the `wrapper.conf` file:

1. Open the Sumo Logic `wrapper.conf` file.
2. Under Application Parameters, add `wrapper.app.parameter.x=--clobber` to the first open line:

```
# Application parameters.  
wrapper.app.parameter.1=com.sumologic.scala.collector.Collector  
wrapper.app.parameter.2=-b  
wrapper.app.parameter.3=installerSources/selected.json  
wrapper.app.parameter.4=--clobber
```

3. Start the Collector.

To set the clobber flag in a `sumo.conf` file:

1. Create a file named `/etc/sumo.conf` (Mac or Linux) or `C:\sumo\sumo.conf` (Windows).
2. Type `clobber=true` in the file, then save and close it.
3. Start the Collector.

Setting a Collector as Ephemeral

During the installation process, if a Collector is flagged as ephemeral, the Collector will be deleted automatically after being offline for 12 hours. This can be helpful when using certain APIs, for example, where Amazon Machine Images (AMIs) are constantly created as new Collectors, but serve a purpose for only a short while. After being offline for a while, the AMI-created Collector is automatically deleted.



A Collector can only be set as ephemeral only before it has been registered (activated) with Sumo Logic. The option cannot be specified for existing Collectors.

You can choose to either modify the `wrapper.conf` file or the `sumo.conf` file to pass the ephemeral flag to Sumo Logic; using `sumo.conf` is the preferred method.

To set the ephemeral flag in a `sumo.conf` file:

1. If you haven't already created `sumo.conf`, create a file named `sumo.conf` (in the `/etc/sumo.conf` directory on Mac or Linux; in the `C:\sumo` directory on Windows).
2. Type `ephemeral=true` in the file, then save and close it.
3. Start the Collector.

To set the ephemeral flag in the `wrapper.conf` file:

1. Open the Sumo Logic `wrapper.conf` file.
2. Under Application Parameters, add `wrapper.app.parameter.x=--ephemeral` to the first open line:

```
# Application parameters.  
wrapper.app.parameter.1=com.sumologic.scala.collector.Collector  
wrapper.app.parameter.2=-b  
wrapper.app.parameter.3=installerSources/selected.json  
wrapper.app.parameter.4=--ephemeral
```

3. Start the Collector.

What happens to logs collected from ephemeral Collectors?

Logs are only deleted from Sumo Logic when your organization's retention period expires, including data from ephemeral Collectors that have been deleted, meaning that you'll still be able to run searches against logs collected from deleted Collectors. However, after deleting a Collector, the internal mapping of the Collector name and Source names to their internal IDs are broken, so that the `"_collector"` and `"_source"` metadata fields are no longer available when searching logs submitted by the deleted Collector. All other metadata fields such as `_sourceCategory`, `_sourceHost` and `_sourceName` will continue to work for targeting logs previously sent from a deleted Collector.

Configuring an Installed Collector behind a proxy

If you install a Collector on a machine in your deployment that doesn't have access to the internet, so a proxy is used to relay data to Sumo Logic. Sumo Logic supports **authentication proxies** as well as **NTLM proxies**.

To connect to Sumo Logic via a proxy you'll need to edit the `wrapper.conf` file by adding certain properties. You can find the `wrapper.conf` file in the installation directory.

Important: Proxy settings must be configured prior to starting and registering the Collector. The Collector can't be authenticated without connecting to the Sumo Logic service.

Connecting via an authentication proxy

To connect to Sumo Logic via a proxy you'll need to edit the `sumo.conf` file with the proxy IP, port, user name, and password properties. You can find the `sumo.conf` file in the installation directory.

To connect an Installed Collector to Sumo Logic via an authentication proxy:

1. Open the `wrapper.conf` file.
2. Under **Wrapper Java Properties**, add the following properties:

```
# Java Additional Parameters  
wrapper.java.additional.1=-XX:+UseParallelGC  
wrapper.java.additional.2=-server  
wrapper.java.additional.3=-Dhttps.proxyHost=host.mydomain.com  
wrapper.java.additional.4=-Dhttps.proxyPort=8080  
wrapper.java.additional.5=-Dhttps.proxyUser=myUserName  
wrapper.java.additional.6=-Dhttps.proxyPassword=pwd01234
```

Use the next available number in the `wrapper.java.additional` category in your conf file (in our example we used 3 through 6).

Connecting via an NTLM proxy

NTLM is a proprietary Microsoft authentication scheme that's optimized for Windows operating systems. In addition to the properties used by authentication proxies, you'll need to add the domain of the Collector to `sumo.conf`.

Additionally, to allow the Collector to authenticate every time it's restarted, you'll need to keep the `sumo.conf` in the installation directory, along with all the NTLM properties.

To connect an Installed Collector to Sumo Logic via an NTLM proxy:

1. Open the `wrapper.conf` file.
2. Under **Wrapper Java Properties**, add the following properties:

```
# Java Additional Parameters
wrapper.java.additional.1=-XX:+UseParallelGC
wrapper.java.additional.2=-server
wrapper.java.additional.3=-Dhttps.proxyHost=host.mydomain.com
wrapper.java.additional.4=-Dhttps.proxyPort=80
wrapper.java.additional.5=-Dhttps.proxyUser=myUserName
wrapper.java.additional.6=-Dhttps.proxyPassword=pwd01234
wrapper.java.additional.7=-Dhttps.proxyNtlmDomain=ES
```

Use the next available number in the `wrapper.java.additional` category in your conf file (in our example we used 3 through 7).

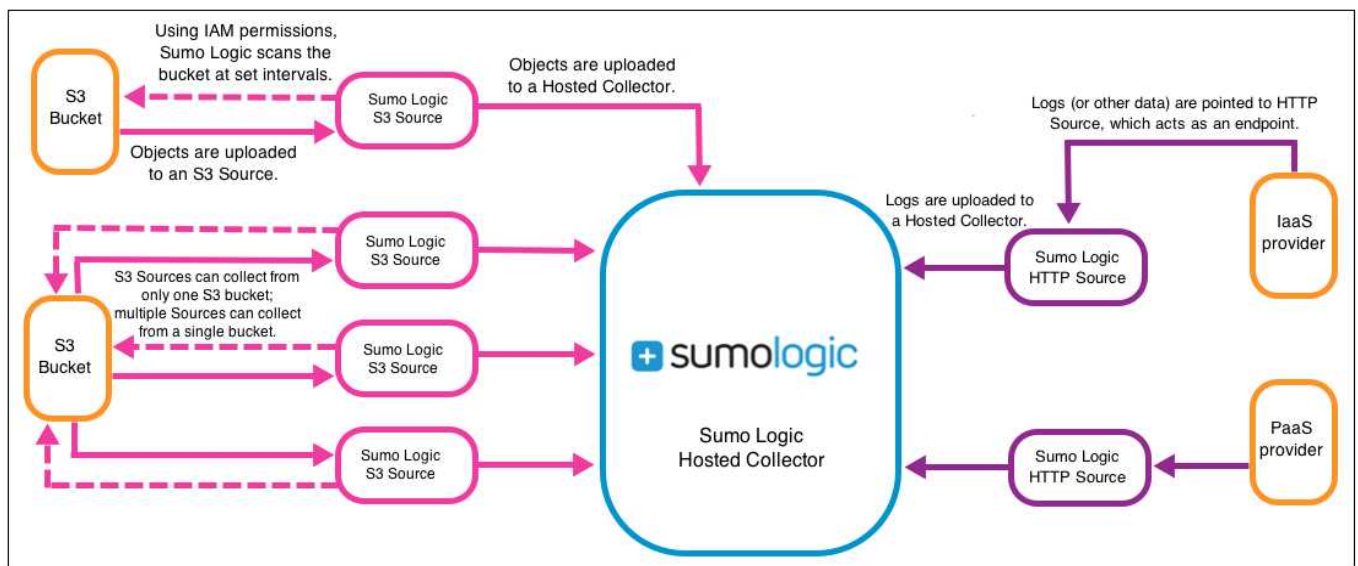
Hosted Collectors

With Hosted Collectors and S3 Sources, data stored in your organization's S3 bucket can be uploaded into Sumo Logic, allowing you to perform the same searches and forensics on data no matter where it's stored. Additionally, [HTTP Sources](#) can be configured to upload data to Sumo Logic.

Just as Local Collectors, you can monitor activity of Hosted Collectors using the Status tab of the Sumo Logic Web Application.

Setting up a Hosted Collector

A **Hosted Collector** is not installed on a local system in your deployment. Instead, Sumo Logic hosts the Collector (along with a Collector's Sources) in AWS. With a Hosted Collector, you can configure S3 Sources, allowing you to move data from your S3 bucket directly in to Sumo Logic, or HTTP Sources, that you'll use to upload data through a secure URL. A single Hosted Collector can be configured with any number of S3 and HTTP Sources.



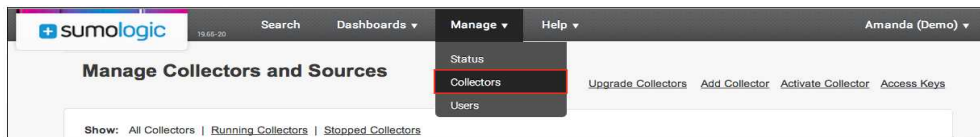
Note: Until you configure an S3 or HTTP Source no data will be uploaded via the Hosted Collector to Sumo Logic. In order to add an S3 Source, you'll need to have IAM in your organization's AWS account to grant Sumo Logic access to your bucket. For more information, see [Granting Access to an S3 bucket](#). There are no prerequisites for HTTP Source configuration.

To set up a Hosted Collector:

1. In the Sumo Logic Web Application select **Manage > Collectors**.



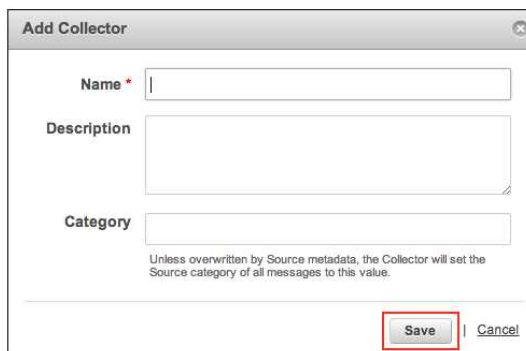
2. Click **Add Collector**.



3. Click **Hosted Collector**.



4. In the **Add Collector** dialog box, type a **Name** for the Collector as well as an optional Description and Category. Then click **Save**.



After the Collector has been set up, it appears in the Collectors tab as a Hosted Collector:

Type	Status	Source Category	Sources	Last Hour
Hosted	✓	awss3	1	None
Local	✓	auth_logs	4	None

Step 3. Add Sources

In order to upload data to Sumo Logic, you'll need to add one or more Sources for the Collectors you've already configured.

It's easy to add more Sources later from the Collectors tab of the Web Application. Each Source is configured to collect files in a specific way, depending on the type of Collector you'll configure the Source for.

Installed Collectors can use any of the following Source types:

- [Local File Source](#)
- [Remote File Source](#)
- [Syslog Source](#)
- [Local Windows Event Log Source](#)
- [Remote Windows Event Log Source](#)
- [Script Source](#)
- [Script Action](#)

Hosted Collectors can use:

- [Amazon S3 Source](#)
- [HTTP Source](#)

Configuring a Local File Source

To collect log messages from the same machine where a Collector is installed, create a Local File Source. If you are editing a Source, metadata changes are reflected going forward. Metadata for previously collected logs will not be retroactively changed.

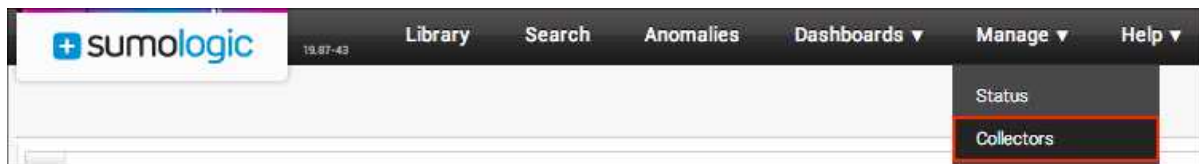
Log files collected via a Local File Source must be encoded in UTF-8 or ASCII.



Want to collect local Windows Event Log data? You can set up a [local Windows Event Log Source](#) to collect those logs.

To configure a Local File Source:

1. In the Web Application, select **Manage > Collectors**.



2. Find the name of the installed Collector to which you'd like to add a Source. Click **Add...** then choose **Add Source** from the pop-up menu.



3. Select **Local File** for the Source type.



4. Set the following choices:

Name* LocalFileSource

Description

File Path* /var/log/**
 Absolute path expression to one or more files, or Windows UNC Share path.
 For example: /var/log/messages or /var/log/*.log or /var/log/**

Collection should begin 7 days ago
 (starts approx. at 11/15/2013 10AM)

Source Host Local_Server
 Host name for the local machine, e.g. LDAP_Server

Source Category OS_Security
 Log category metadata to use later for querying, e.g. OS_Security

Name. Type the name you'd like to display for the new Source. Description is optional.

File Path. List the full path to the file you want to collect. For files on Windows systems (not including Windows Events), enter the absolute path including the drive letter. Escape special characters and spaces with a backslash (\). If you are collecting from Windows using CIFS/SMB, see Collecting from a

Windows UNC Share Path.



Use a single asterisk wildcard [*] for file or folder names [var/foo/*.log]. Use two asterisks [**] to recurse within directories and subdirectories [var/**/*.log].

Collection start time. Choose how far back you'd like to begin collecting historical logs. For example, choose 7 days ago to begin collection with logs generated seven days ago. To begin more recently, choose 24 hours.

Source Host. Sumo Logic uses the hostname assigned by the OS unless you type a different host name. Hostname metadata is stored in a searchable field called `_sourceHost`. Avoid using spaces in metadata tags so that you do not have to quote the source host or the source category in the search query field. For more information, see Establishing Metadata Conventions.

Source Category. Enter any string to tag the output collected from this Source. (Category metadata is stored in a searchable field called `_sourceCategory`.)

- Set any of the following options under **Advanced**:

Advanced

Blacklist
 One or more comma separated path expressions describing the files to be excluded.
 For example: /var/log/**/*.bak, /var/oldlog/*.log

Enable Timestamp Parsing ☒ Extract timestamp information from log file entries

Time Zone
☐ Use time zone from log file. If none is present use:
 Select a time zone
☒ Ignore time zone from log file and instead use:
 Select a time zone

Timestamp Format
☒ Automatically detect the format ☐ Specify a format

Enable Multiline Processing ☐ Detect messages spanning multiple lines
☒ Infer Boundaries - Detect message boundaries automatically
 Please note, Infer Boundaries may not be accurate for all log types.
☐ Boundary Regex - Expression to match message boundary e.g. (?<\\n)(r+)

Blacklist. In the Blacklist field, enter the path for files to exclude from the Source collection. Wildcard syntax is allowed when specifying unwanted files. For example, you are collecting `/var/log/*.log` but don't want to collect `unwanted*.log`, then specify `/var/log/unwanted*.log` in the blacklist. You can also exclude subdirectories, for example, if you are collecting `/var/log/**/*.log` but do not want to collect anything from `/var/log/unwanted` directory, specify `/var/log/unwanted`.



You don't need to blacklist compressed files. Sumo Logic automatically excludes compressed files when collecting data.

Enable Timestamp Parsing. This option is selected by default. If it's deselected, no timestamp information is parsed at all.

Time Zone. There are two options for Time Zone. You can use the time zone present in your log files, and then choose an option in case time zone information is missing from a log message. Or, you can have Sumo Logic completely disregard any time zone information present in logs by forcing a time zone. It's very important to have the proper time zone set, no matter which option you choose. If the time zone of logs can't be determined, Sumo Logic assigns logs UTC; if the rest of your logs are from another time zone your search results will be affected.

Timestamp Format. By default, Sumo Logic will automatically detect the timestamp format of your logs. However, you can manually specify a timestamp format for a Source. See [Specifying a Timestamp Format](#) for more information.

Multiline Processing. Deselect **Multiline Processing** to avoid unnecessary processing if you are collecting single-message-per-line files (for example, Linux system.log). If you're working with multi-line messages (for example, log4J or exception stack traces) then select this option.

Infer Boundaries. Enable when you want Sumo Logic to automatically attempt to determine which lines belong to the same message.



If you deselect the Infer Boundaries option, you will need to enter a regular expression in the Boundary Regex field to use for detecting the entire first line of multi-line messages.

Boundary Regex. You can specify the boundary between messages using a regular expression. Enter a regular expression for the full first line of every multi-line message in your log files. For an example, see [Boundary Regex](#).

6. [Create any filters](#) you'd like for the new Source.
7. When you are finished configuring the Source click **Save**.

You can return to this dialog and edit the settings for the Source at any time.

How does Sumo Logic handle log file rotation?

Sumo Logic handles log file rotation without any additional configuration. For example, let's say that an active log file is named error.log, and that it's rotated to error.log.timestamp every night. In this case, Sumo Logic detects that the file is rotated, and continues to monitor both the rotated file as well as the new error.log file, assuming that the first 2048 bytes of the error.log file and the rotated file are different.

Configuring a Remote File Source

Remote file tail data is collected via SSH. If you are collecting from a Windows machine, please see the note following on Prerequisite for Windows. If you are editing a Source, metadata changes are reflected going forward. Metadata for previously collected log data will not be retroactively changed.

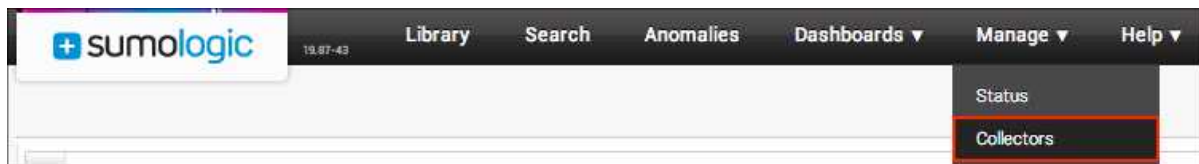
Log files collected via a Remote File Source must be encoded in UTF-8 or ASCII.



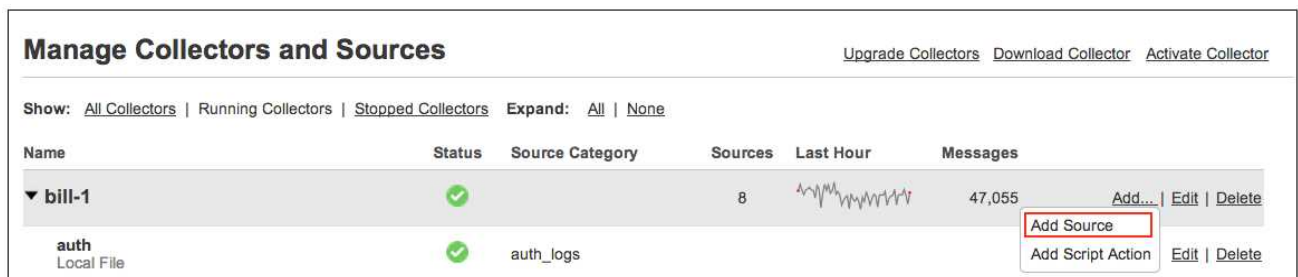
To collect remote Windows logs using CIFS/SMB, see [Collecting from a Windows UNC Share Path](#). To collect Windows Events, see [Configuring a Windows Event Log Source](#).

To configure a Remote File Source:

1. In the Web Application, select **Manage > Collectors**.



2. Find the name of the installed Collector to which you'd like to add a Source. Click **Add...** then choose **Add Source** from the pop-up menu.



3. Select **Remote File** for the Source type.



4. Set the following:

Name. Type the name you'd like to display for the new Source. **Description** is optional. Source name metadata is stored in a searchable field called `_sourceName`.

Host. Enter the hostname or the IP address of the remote machine (the hostname entered must be the system hostname or IP address and cannot be changed). The hostname is stored in a searchable field called `_sourceHost`.

Port. If your SSH server is listening on a nonstandard port, type the port number.

Path Expression. Enter the absolute path expression to the file the Source should tail. Remote File Sources support wildcards in file paths. If the timestamp formats for the files are not identical, set up a separate Remote File Source for each file.



For Windows collections using [Open SSH](#) and [Cygwin](#), specify the File path starting with `/cygdrive`. For example, if the path is `"C:\mandy test\6.log"` enter `"/cygdrive/c/mandy\ test/6.log"` in the File field. Use `"\"` to escape any spaces if they are present in the file path.

Source Category. Type any string to tag the output collected from this Source with searchable metadata. For example, type `firewall` to tag all entries from this Source in a field called `_sourceCategory`. Type `_sourceCategory=firewall` in the Search field to return results from this Source. For more information, see [Establishing Metadata Conventions](#).

5. Choose the type of **Credentials** used for this Source:

- **Username and Password.** Enter valid user credentials for the remote machine.
- **Local SSH Config.** Enter the username and the absolute path, including file name, to the PEM SSH key file located on the Collector host. Enter a password if required.

3. Set any of the following under **Advanced**:

Blacklist. Optional. Add any files to be excluded by including one or more path expressions separated by commas. Note that this field takes a maximum of 10240 characters.

Enable Timestamp Parsing. This option is selected by default. If it's deselected, no timestamp information is parsed at all.

Time Zone. There are two options for Time Zone. You can use the time zone present in your log files, and then choose an option in case time zone information is missing from a log message. Or, you can have Sumo Logic completely disregard any time zone information present in logs by forcing a time zone. It's very important to have the proper time zone set, no matter which option you choose. If the time zone of logs can't be determined, Sumo Logic assigns logs UTC; if the rest of your logs are from another time zone your search results will be affected.

Timestamp Format. By default, Sumo Logic will automatically detect the timestamp format of your logs. However, you can manually specify a timestamp format for a Source. See [Specifying a Timestamp Format](#) for more information.

Multiline Processing. Deselect **Multiline Processing** to avoid unnecessary processing if you are collecting single-message-per-line files (for example, Linux system.log). If you're working with multi-line messages (for example, log4J or exception stack traces) then select this option.

Infer Boundaries. Enable when you want Sumo Logic to automatically attempt to determine which lines belong to the same message.



If you deselect the Infer Boundaries option, you will need to enter a regular expression in the Boundary Regex field to use for detecting the entire first line of multi-line messages.

Boundary Regex. You can specify the boundary between messages using a regular expression. Enter a regular expression for the full first line of every multi-line message in your log files. For an example, see [Boundary Regex](#).

7. [Create any filters](#) you'd like for the new Source.
8. When you are finished configuring the Source click **Save**.

Configuring a Local Windows Events Source

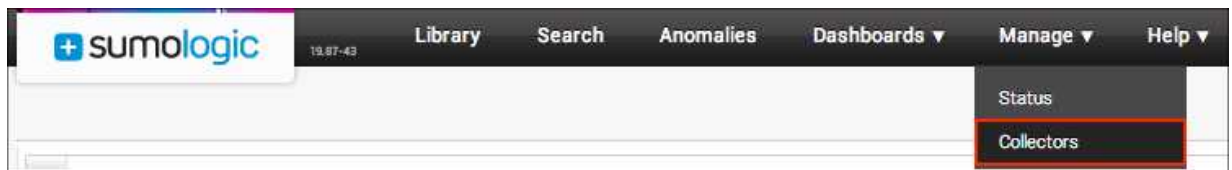
Set up a Local Windows Event Log Source to collect local events you would normally see in the Windows Event Viewer. Setting up a Local Windows Event Source is a quick process. There are no prerequisites for setting up the Source, and you'll begin collecting logs within a minute or so.

Local Windows Events Sources can only be configured on systems running Windows Server 2008 and later.

Local Windows Event Log Sources are only for collecting Windows Event Logs. All other types of log sources need to be configured either as a Remote File Source or as a Local File Source.

To configure a Local Windows Event Log Source:

1. In the Web Application, select **Manage > Collectors**.



2. Find the name of the installed Collector to which you'd like to add a Source. Click **Add...** then choose **Add Source** from the pop-up menu.



3. Click **Windows Event Log**.



4. Choose **Local** for the Type of Windows Event Log Source.



5. Set the following:

Name. Type the name you'd like to display for the new Source. **Description** is optional.

Source Category. Enter a string used to tag the output collected from this Source with searchable metadata. For example, typing `web_apps` tags all the logs from this Source in the `sourceCategory` field, so running a search on `_sourceCategory=web_apps` would return logs from this Source. For more information, see [Establishing metadata conventions](#).

Windows Event Types. Select the Event Types you want to collect, or choose **All Events** to select all options.

The screenshot shows a configuration dialog for a Windows Event Log Source. It is divided into several sections:

- Name:** A text field containing "LocalWinEvents Source".
- Description:** An empty text field.
- Windows host(s):** An empty text field with a tooltip that says "List hosts, separated by commas. If left blank, localhost is assumed."
- Source Category:** A text field containing "OS_Security" with a tooltip that says "Log category metadata to use later for querying, e.g. OS_Security".
- Windows Domain:** An empty text field.
- Username:** An empty text field.
- Password:** An empty text field.
- Windows Event Types:** A section with checkboxes for "All Events", "Security", "System", "Application", and "Others". All five checkboxes are currently checked.
- Collection should begin:** A dropdown menu showing "24 hours ago" with a tooltip that says "(starts approx. at 12/2/2013 1PM)".

- Under **Advanced**, choose an option for **Time Zone**. By default, Local Windows Event Log Sources determines the time zone of logs from the timestamps present in log messages. To over-ride this default, choose **Ignore time zone from log files** and then choose a time zone, such as UTC, from the drop-down menu.
- [Create any filters](#) you'd like for the new Source.
- When you are finished configuring the Source click **Save**.

You can return to this dialog and edit the settings for the Source at any time.

Configuring a Remote Windows Event Log Source

Set up a Remote Windows Event Log Source to collect remote events you would normally see in the Windows Event Viewer. Before collecting Windows Events from a remote machine, you'll need to configure a few settings.

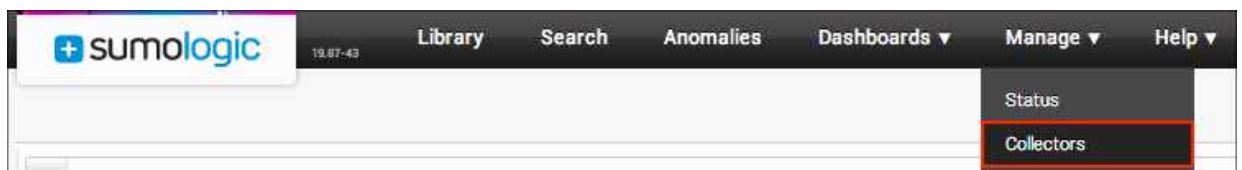
A Windows Event Log Source collects the unique formats of Windows Events using the WMI (Windows Management Instrumentation) interface. You can configure a Windows Event Log Source to collect from multiple remote machines by designating a comma-separated list of remote hostnames.



Prerequisites: In order to collect remote Windows Events, you will first need to configure a domain user and adjust firewall and RPC settings. See [Collecting Windows Events](#) for more information.

To configure a remote Windows Event Log Source:

1. Complete the prerequisites to collecting remote events.
2. In the Web Application, select **Manage > Collectors**.



3. Find the name of the installed Collector to which you'd like to add a Source. Click **Add...** then choose **Add Source** from the pop-up menu.



4. Select the **Windows Event Log** Source.



5. Choose **Remote** for the Type of Windows Event Log Source.



6. Set the following:

The screenshot shows a configuration dialog for a 'Win Event Remote Source'. The fields are as follows:

- Name***: Win Event Remote Source
- Description**: (empty)
- Windows host(s)**: (empty). Below the field is the text: 'List hosts, separated by commas. If left blank, localhost is assumed.'
- Source Category**: OS_Security. Below the field is the text: 'Log category metadata to use later for querying, e.g. OS_Security'
- Windows Domain***: SLRemote
- Username***: username
- Password***: (masked with dots)
- Windows Event Types***: A group of checkboxes including 'All Events' (checked), 'Security' (checked), 'System' (checked), 'Application' (checked), and 'Others' (checked).
- Collection should begin**: 24 hours ago. Below this is the text: '(starts approx. at 12/2/2013 10AM)'

Name. Type the name you'd like to display for this Source in the Sumo Logic Web Application. **Description** is optional.

Windows host(s). Enter one or more hostnames for the Windows machines from which you want to collect Windows Events. If you'd like to collect from more than one remote host, separate the hostnames with a comma. (If you enter more than one hostname, each host must have the same domain user. See [Collecting Windows Events](#) for more

information.)

Source Category. Enter a string used to tag the output collected from this Source with searchable metadata. For example, typing `web_apps` tags all the logs from this Source in the `sourceCategory` field. For more information, see [Establishing metadata conventions](#).

Windows Domain. Type the name of the Windows Domain, the **Username** for this host, and the **Password**.

Windows Event Types. Select the Event types you want to collect. Choose **All Events** to select all options.

Collection start time. Choose how far back you'd like to begin collecting historical logs. For example, choose 7 days ago to begin collection with logs generated seven days ago. To begin more recently, choose 24 hours.

7. Set any of the following under **Advanced**:

Time Zone. There are two options for Time Zone. You can use the time zone present in your log files, and then choose an option in case time zone information is missing from a log message. Or, you can have Sumo Logic completely disregard any time zone information present in logs by forcing a time zone. It's very important to have the proper time zone set, no matter which option you choose. If the time zone of logs can't be determined, Sumo Logic assigns logs UTC; if the rest of your logs are from another time zone your search results will be affected.

Timestamp Format. By default, Sumo Logic will automatically detect the timestamp format of your logs. However, you can manually specify a timestamp format for a Source. See [Specifying a Timestamp Format](#) for more information.

8. [Create any filters](#) you'd like for the new Source.

9. When you are finished configuring the Source click **Save**.

You can return to this dialog and edit the settings for the Source at any time.

Prerequisites for Windows Collections

Sumo Logic requires a few extra steps when you set up collections in a Windows environment:

- To collect Windows Events, first follow the instructions for setting up a domain user and configuring firewall and RPC settings. For detailed instructions, see [Collecting Windows Events](#).
- For remote file collections from Windows systems, choose one of these two methods:
 - [Set up a UNC Share Path to collect Windows logs using CIFS/SMB](#)
 - [Set up a third-party tool on the target system to handle SSH](#)

Collecting Windows Logs from a UNC Share Path

As an alternative to using SSH for remote Windows collections, Sumo Logic Collectors can collect files remotely using CIFS/SMB by configuring a Local File Source (not a Remote File Source) with a UNC share path.

Here is an overview of the required steps:

- [Install the Sumo Logic Collector](#) for Windows.
- On the machine where the files reside (the target or remote machine), use Windows Advanced Sharing options to create a UNC share for the log directory.
- [Set up a Local File Source](#)



The Collector must reside within the same Active Directory domain as the target host, and the target host must allow access without requiring a password.

Step 1. Install a Sumo Logic Collector.

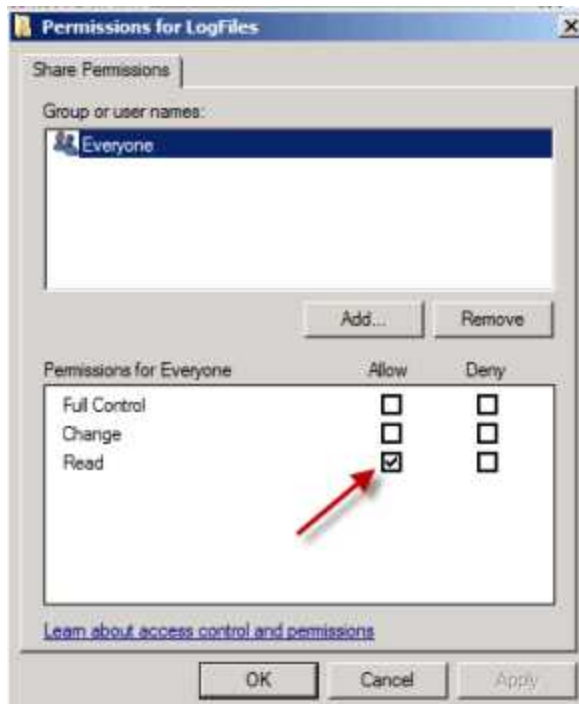
- [Install the Sumo Logic Collector](#) on a machine within the same Active Directory domain as the target system where files reside.

Step 2. Set UNC share permissions.

1. Set up the UNC share permissions (Share with "Everyone" and "Read-Only") for the folder on the target machine.
 - Open Explorer in the machine where the files reside.
 - Right-click the log directory, and select **Properties**.
 - Click **Advanced Sharing**.
 - In the Advanced Sharing dialog, give the log directory a share name (or just use the actual folder name), and then click **Permissions**.
 - Set the permissions for **Everyone to Read**.

NOTE: The Collector operates as the SYSTEM user, and cannot utilize drive mappings; you can't log in to the Collector and map **F: to \\server\share\path**. The Collector runs as SYSTEM, and SYSTEM is not you, nor is SYSTEM logged on, so SYSTEM can't use drive maps. Setting up the share for only a specific user prevents the Collector from successfully accessing the directory. Make

sure you set permissions to "Everyone".



2. Click OK. When the Permissions dialog closes, you will see your UNC path listed under Network Path. This is the exact path you will enter when you are configuring a Local File Source in the Sumo Logic Web Application.



3. Verify that you have set up the share correctly. Open Explorer on the machine where the Collector is installed. Type in the UNC share path. If you can see the log files listed, you can collect them. If Explorer asks you to enter a password, then you have not set up permissions correctly. Make sure that permissions for the folder are set to "Everyone" and "Read-Only."
4. From the Sumo Logic Web Application, create a new Local File Source.

5. Enter the file path to the UNC share. For this example, the UNC path looks like this:

```
\\WIN-QR0406514NE\c$\LogFiles\*
```

In general, a UNC path has this format:

```
\\server\share\file_path
```

6. The **server** portion of a UNC path references the server name set by a system administrator, or an IP address. The **share** portion of a UNC name references a labeled share point created by an administrator, as in Step 2. The **file path** portion of a UNC name references the local subdirectories beneath the share point.
7. Save your Local File Source configuration. Wait a few seconds, and then click the **Status** tab to check the message volume for the Collector.

Using a third-party client to handle SSH in Windows

There are two options for collecting Windows logs remotely:

- Set up a Local File Source to collect (remote files) via CIFS/SMB. [Learn more](#).
- Set up a Remote File Source to collect via SSH. This topic describes steps to enable SSH collection.

Windows does not handle SSH natively. You will need to install a third-party product (OpenSSH) to enable this type of collection.

To install OpenSSH and Cygwin:

1. [Download OpenSSH from Sourceforge](#).
2. Install OpenSSH. to C:\OpenSSH or another directory.
3. [Download and install Cygwin](#).
4. Open a cmd window and start the SSH service: run "**net start opensshd**".
5. SSH the window system. Verify that SSH works and that you can tail a file. For example, for a user called "mandy" run command in terminal:

```
ssh mandy@192.168.1.114
(enter password)
tail -f -n+1 /cygdrive/c/mandy\ test/6.log
```

When you [configure the Remote File Source](#) to collect from the Windows machine, make sure to:

1. Specify the host as the Windows system.
2. Specify the File path starting with **/cygdrive**. For example, enter **"/cygdrive/c/mandy\ test/6.log"** in the File field if the path is **"C:\mandy test\6.log"**.



Use **"\"** to escape any spaces if they are present in the file path.

Configuring a Local Windows Performance Monitor Log Source

Set up a Local Windows Performance Monitor Log Source to collect performance data that you would normally see in the Windows Performance Monitor. Setting up a Local Windows Performance Monitor Source is a quick process. There are no prerequisites for setting up the Source, and you'll begin collecting logs within a minute or so.

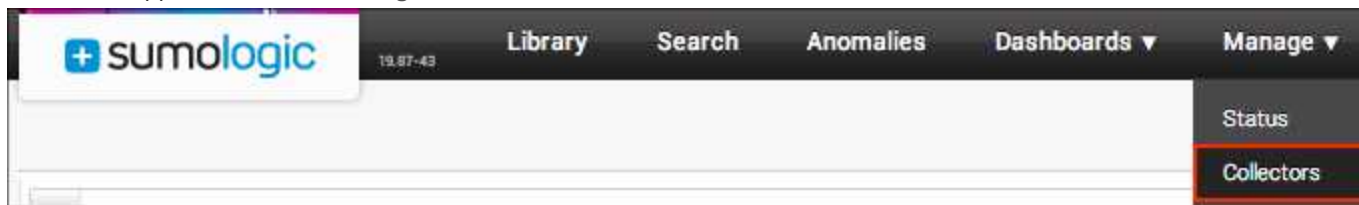


Local Windows Performance Monitor Log Sources can only be configured on systems running Windows Server 2008 and later.

Windows Performance Monitor Sources use the WMI Query Language (WQL) to collect data at a frequency you choose. To learn more, see [Querying with WQL](#) at MSDN.

To configure a Local Windows Performance Monitor Log Source:

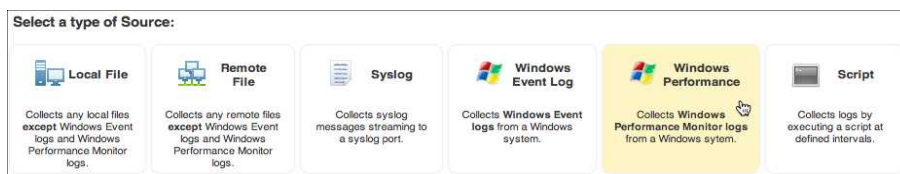
1. In the Web Application, select **Manage > Collectors**.



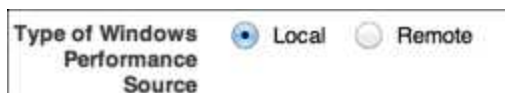
2. Find the name of the installed Collector to which you'd like to add a Source. Click **Add...** then choose **Add Source** from the menu.



3. Click **Windows Performance**.



4. Choose **Local** for the Type of Windows Performance Source.



5. Set the following:

The screenshot shows a form for configuring a new Source. It includes the following fields and options:

- Name:** A text input field with a warning icon and a note: "Maximum name length is 128 characters".
- Description:** A text input field.
- Windows host(s):** A text input field with a note: "List hosts, separated by commas. If left blank, localhost is assumed."
- Frequency:** A dropdown menu currently set to "Every 15 Minutes".
- Source Category:** A text input field with a note: "Log category metadata to use later for querying, e.g. OS_Security".

- **Name.** Type the name to display for the new Source. **Description** is optional.
 - **Frequency.** Depending on your Windows system and its needs, select a frequency to run the selected queries. If your Windows system is relatively stable, a frequency of 15m should be appropriate. (Selecting a frequency of 1m could flood your system with logs and create an undesirable outcome.)
 - **Source Category.** Enter a string used to tag the output collected from this Source with searchable metadata. For example, typing `web_apps` tags all the logs from this Source in the `sourceCategory` field, so running a search on `_sourceCategory=web_apps` would return logs from this Source. For more information, see [Establishing metadata conventions](#).
6. **Filters.** (Optional.) To add filters for the new source, click **Add Filter**. Enter a name, a filter, and select the type. Then click **Apply**.
7. **Perfmon Queries.** Select from the provided default Perfom Queries, or create your own custom query.

The screenshot shows the "Perfmon Queries" section. It contains a table of default queries and a form to add a custom query.

Name	Query
<input checked="" type="checkbox"/> CPU	select * from Win32_PerfFormattedData_PerfOS_Processor
<input type="checkbox"/> Logical Disk	select * from Win32_PerfFormattedData_PerfDisk_LogicalDisk
<input type="checkbox"/> Physical Disk	select * from Win32_PerfFormattedData_PerfDisk_PhysicalDisk
<input type="checkbox"/> Memory	select * from Win32_PerfFormattedData_PerfOS_Memory
<input type="checkbox"/> Network	select * from Win32_PerfFormattedData_Tcpip_NetworkInterface

Below the table is a form to add a custom query:

Name

Query

Enter a query in the format of select X from Y.

|

[Add Query](#)

- Click the query's check box to select it.
- To add a custom query, click **Add Query**, enter a name and the query. Then click **Add**.

8. Set the following:

Name* 
Maximum name length is 128 characters

Description

Windows host(s)
List hosts, separated by commas. If left blank, localhost is assumed.

Frequency

Source Category
Log category metadata to use later for querying, e.g. OS_Security

- **Name.** Type the name to display for the new Source. **Description** is optional.
 - **Frequency.** Depending on your Windows system and its needs, select a frequency to run the selected queries. If your Windows system is relatively stable, a frequency of 15m should be appropriate. (Selecting a frequency of 1m could flood your system with logs and create an undesirable outcome.)
 - **Source Category.** Enter a string used to tag the output collected from this Source with searchable metadata. For example, typing **web_apps** tags all the logs from this Source in the sourceCategory field, so running a search on _sourceCategory=web_apps would return logs from this Source. For more information, see Establishing metadata conventions.
9. **Filters.** (Optional.) To add filters for the new source, click **Add Filter**. Enter a name, a filter, and select the type. Then click **Apply**.
10. For **Perfmon Queries**, do one of the following:

▼ **Perfmon Queries ***

Name	Query
<input type="checkbox"/> CPU	select * from Win32_PerfFormattedData_PerfOS_Processor
<input type="checkbox"/> Logical Disk	select * from Win32_PerfFormattedData_PerfDisk_LogicalDisk
<input type="checkbox"/> Physical Disk	select * from Win32_PerfFormattedData_PerfDisk_PhysicalDisk
<input type="checkbox"/> Memory	select * from Win32_PerfFormattedData_PerfOS_Memory
<input type="checkbox"/> Network	select * from Win32_PerfFormattedData_Tcpip_NetworkInterface

Name	Query	Add Query
<div> <div>Name</div> <div> <input type="text"/> </div> </div> <div> <div>Query</div> <div> <input type="text"/> </div> </div> <p>Enter a query in the format of select X from Y.</p> <div> <input type="button" value="Add"/> <input type="button" value="Cancel"/> </div>		

- Click the name of a default query's check box to select it.
 - To add a custom query, click **Add Query**, enter a name and the query. Then click **Add**.
11. When you are finished configuring the Source, click **Save**.

You can return to this dialog and edit the settings for the Source at any time.

Configuring a Remote Windows Performance Monitor Log Source

Set up a Remote Windows Performance Monitor Log Source to collect remote performance data you would normally see in the Windows Performance Monitor. Before collecting Windows performance data from a remote machine, you'll need to configure a few settings.

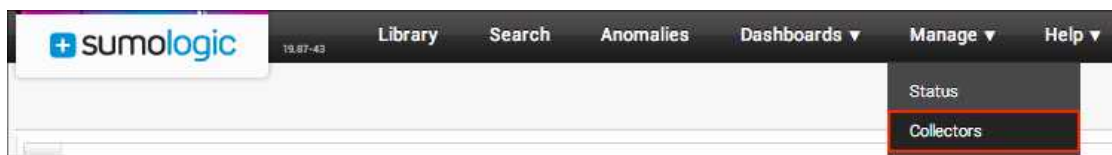
A Windows Performance Monitor Log Source collects the unique formats of Windows Events using the WMI (Windows Management Instrumentation) interface. You can configure a Windows Performance Monitor Log Source to collect from multiple remote machines by designating a comma-separated list of remote hostnames.



Prerequisites: In order to collect remote Windows Events, you will first need to configure a domain user and adjust firewall and RPC settings. See [Collecting Windows Events](#) for more information.

To configure a remote Windows Performance Monitor Log Source:

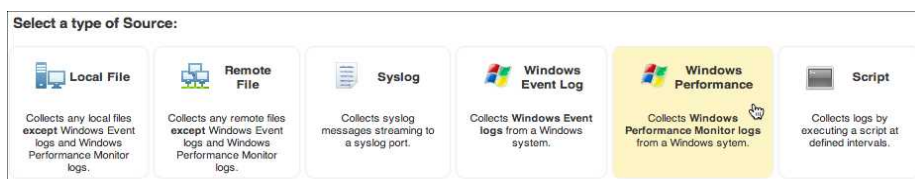
1. Complete the prerequisites to collecting remote events.
2. In the Web Application, select **Manage > Collectors**.



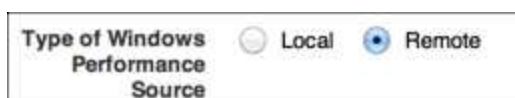
3. Find the name of the installed Collector to which you'd like to add a Source. Click **Add...** then choose **Add Source** from the pop-up menu.



4. Click **Windows Performance**.



5. Choose **Remote** for the **Type of Windows Performance Source**.



6. Set the following:

The screenshot shows a web form for configuring a Sumo Logic source. The form is divided into two main sections. The top section contains fields for 'Name' (with a warning icon and a note 'Maximum name length is 128 characters'), 'Description', 'Windows host(s)' (with a note 'List hosts, separated by commas. If left blank, localhost is assumed.'), 'Frequency' (a dropdown menu currently set to 'Every Minute'), and 'Source Category' (with a note 'Log category metadata to use later for querying, e.g. OS_Security'). The bottom section, enclosed in a rounded rectangle, contains fields for 'Windows Domain', 'Username', and 'Password', each with a red asterisk indicating it is required.

- **Name.** Type the name to display for this Source in the Sumo Logic Web Application.
- **Description** is optional.
- **Windows host(s).** Enter one or more hostnames for the Windows machines from which you want to collect Windows Performance Monitor data. If you'd like to collect from more than one remote host, separate the hostnames with a comma. (If you enter more than one hostname, each host must have the same domain user. See [Collecting Windows Events](#) for more information.)
- **Frequency.** Depending on your Windows system and its needs, select a frequency to run the selected queries. If your Windows system is relatively stable, a frequency of 15m should be appropriate. (Selecting a frequency of 1m could flood your system with logs and create an undesirable outcome.)
- **Source Category.** Enter a string used to tag the output collected from this Source with searchable metadata. For example, typing `web_apps` tags all the logs from this in the `sourceCategory` field. For more information, see [Establishing metadata conventions](#).
- **Windows Domain.** Type the name of the Windows Domain, the Username for this host, and the Password.

7. **Perfmon Queries.** Select from the provided default Perfmon Queries, or create your own custom query.

▼ Perfmon Queries *

Name	Query
<input type="checkbox"/> CPU	select * from Win32_PerfFormattedData_PerfOS_Processor
<input type="checkbox"/> Logical Disk	select * from Win32_PerfFormattedData_PerfDisk_LogicalDisk
<input type="checkbox"/> Physical Disk	select * from Win32_PerfFormattedData_PerfDisk_PhysicalDisk
<input type="checkbox"/> Memory	select * from Win32_PerfFormattedData_PerfOS_Memory
<input type="checkbox"/> Network	select * from Win32_PerfFormattedData_Tcpip_NetworkInterface

Name **Query** [Add Query](#)

Name

Query

Enter a query in the format of select X from Y.

|

- Click the query's check box to select it.
 - To add a custom query, click **Add Query**, enter a name and the query. Then click **Add**.
8. **Filters.** (Optional.) To add filters for the new source, click **Add Filter**. Enter a name, a filter, and select the type. Then click **Apply**.
9. When you are finished configuring the Source, click **Save**.

You can return to this dialog and edit the settings for the Source at any time.

Configuring a Syslog Source

A Sumo Logic Syslog Source operates like a syslog server listening on the designated port to receive syslog messages. Set your syslog-enabled devices to send syslog data to the same port you specify in Sumo Logic Syslog Source configuration.

For multiple syslog collections, set up a separate Source for each and set a separate port number for each.

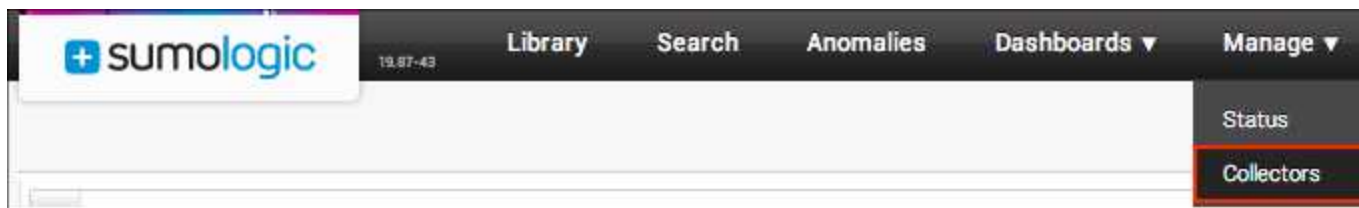


If you are already running a Syslog server, you can either switch to using Sumo Logic Collector as your syslog server (recommended) or you can output the existing syslog server data to a local file, and then set up a Local or a Remote File Source to collect the file.

If you are editing a Source, metadata changes are reflected going forward. Metadata for previously collected log data will not be retroactively changed.

To configure a Syslog Source:

1. In the Web Application, select **Manage > Collectors**.



2. Find the name of the Installed Collector to which you'd like to add a Source. Click **Add...** then choose **Add Source** from the pop-up menu.



3. Select **Syslog** for the Source type.



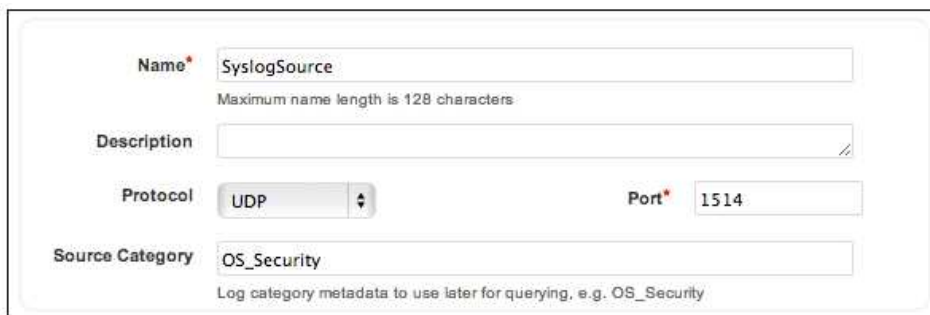
4. Set the following:

Name. Type the name you'd like to display for the new Source. Description is optional. Source name metadata is stored in a searchable field called `_sourceName`.

Protocol. Select the option that your syslog-enabled devices are currently using to send syslog data, either UDP or TCP.

Port. Type the port number for the Source to listen to. If the Collector runs as root (default), use 514. Otherwise, consider 1514 or 5140. Make sure the devices are sending to the same port.

Source Category. Enter any string to tag the output collected from this Source with searchable metadata. For example, enter `firewall` to tag all entries from this Source in a field called `_sourceCategory`. Type `_sourceCategory=firewall` in the Search field to return results from this Source. For more information, see [Establishing Metadata Conventions](#).



The screenshot shows a configuration dialog box for a new Source. It contains the following fields and controls:

- Name:** A text input field containing "SyslogSource". Below it, a small text label says "Maximum name length is 128 characters".
- Description:** A text input field that is currently empty.
- Protocol:** A dropdown menu currently set to "UDP".
- Port:** A text input field containing "1514".
- Source Category:** A text input field containing "OS_Security". Below it, a small text label says "Log category metadata to use later for querying, e.g. OS_Security".

5. Set any of the following under **Advanced**:

Enable Timestamp Parsing. This option is selected by default. If it's deselected, no timestamp information is parsed at all.

Time Zone. There are two options for Time Zone. You can use the time zone present in your log files, and then choose an option in case time zone information is missing from a log message. Or, you can have Sumo Logic completely disregard any time zone information present in logs by forcing a time zone. It's very important to have the proper time zone set, no matter which option you choose. If the time zone of logs can't be determined, Sumo Logic assigns logs UTC; if the rest of your logs are from another time zone your search results will be affected.

Timestamp Format. By default, Sumo Logic will automatically detect the timestamp format of your logs. However, you can manually specify a timestamp format for a Source. See [Specifying a Timestamp Format](#) for more information.

9. Create any filters you'd like for the new Source.
10. When you are finished configuring the Source click **Save**.

You can return to this dialog and edit the settings for the Source at any time.

Configuring a Script Source

If you need to collect data that isn't stored in log files (like system performance metrics, database records, or perhaps data output from third-party monitoring solutions) you can use a Script Source that uses a script to fetch those custom sources of data. The script executes at defined intervals and then sends the data to Sumo Logic for analysis. A Script Source allows you to collect all sorts of data from any supported OS, including data from command-line tools (for example, as iostat) or transient or unstable data sources.



Once a Script Source is configured, access to the machine running the Collector associated with the Source is granted to all Sumo Logic users with roles that include Collector management.

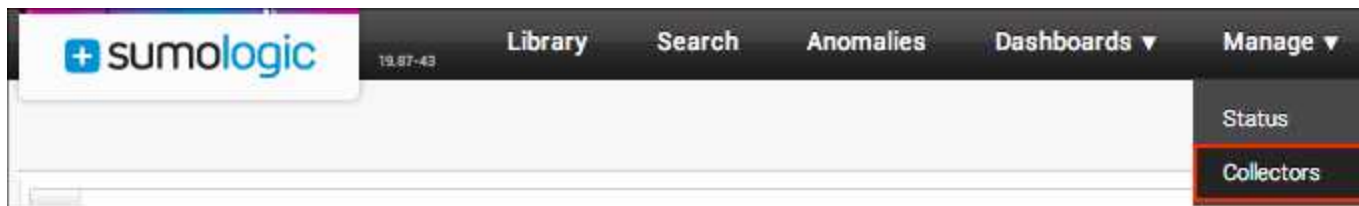
Preparing your script

Collecting from a Script Source depends on a well-constructed script. When considering the data you'd like to collect through a script, keep the following in mind:

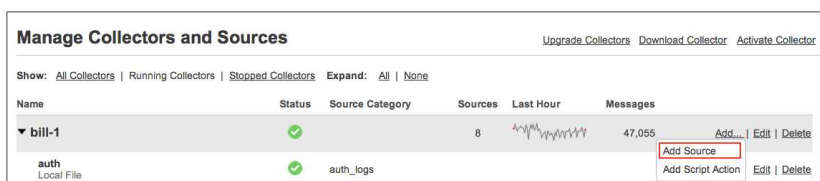
- The script must run on the host computer; remote scripts won't result in data collection. The Script Source assumes that the Collector is running the host where the script is executed. However, the script itself can connect to remote hosts to gather relevant information.
- Supported script types include bat (Windows only), Visual Basic (Windows only), Python, Perl, and bash.
- Wildcards are not supported in these scripts.

To configure a Script Source:

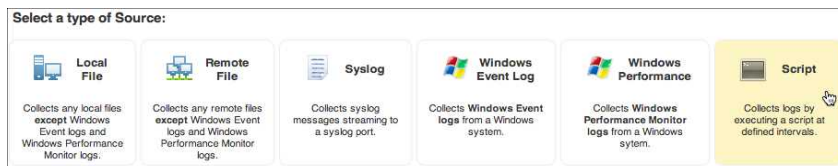
1. In the Web Application, select **Manage > Collectors**.



2. Find the name of the installed Collector to which you'd like to add a Source. Click **Add...** then choose **Add Source** from the pop-up menu.



3. Select **Script** for the Source type.



4. Enter a **Name** to display for the new Source. Description is optional. Source name metadata is stored in a searchable field called **_sourceName**.
5. For **Source Host**, enter the hostname or the IP address of the machine. The hostname is stored in a searchable field called **_sourceHost**
6. For **Source Category**, enter any information you'd like to include in the metadata.
7. For **Frequency**, do one of the following:
 - Choose an option to run the script at the selected frequency.
 - Choose **Other (CRON Expression)** if you'd like to set a customized frequency using a CRON Expression, then type the CRON Expression in the Expression text box.



Using a CRON Expression allows you to specify an exact time for your script to run, like each day at 2:15 pm, or Monday through Friday at midnight. (Learn more about [supported CRON Expressions](#).)

8. If you'd like to set a timeout for your script, select **Specify a timeout for your command**. If you don't need a timeout, or if you're running a script once daily, we recommend that you leave this option deselected. Learn more [here](#).
9. For **Command**, choose the type of command you're going to use. The options in this menu depend on the type of Collector you're using:



Mac/Linux Command options.



Windows Command options.

10. For **Script**, do one of the following:
 - Choose **Type a path to the script to execute** if you have the script saved to a file location. For example:

Script * ☒ Type a path to the script to execute.

`/Users/username/folder/subFolder/collect-search-interval_2.py`

☐ Type the script to execute.

- **OR**, choose **Type the script to execute** if you'd like to enter the script directly in the Sumo Logic Web Application. Then type the script in the text box. For example:

Script * ☐ Type a path to the script to execute.

☒ Type the script to execute.

`qwinsta.exe`

- For **Working Directory**, you'll only need to enter a path if your script refers to a file indirectly. So, enter the path of the file you'd like to collect if required; otherwise this option can remain blank.
- Under **Advanced** you'll see options regarding timestamps and time zones:
 - **Timestamp Parsing.** By default **Extract timestamp information from log file entries** is selected, meaning that Sumo Logic will use the timestamp information from the data you collect. Deselecting this option turns off all timestamp parsing.
 - **Time Zone.** Select an option under **Use time zone from log file, but if none present use**. Or, if you'd like to override all time zones from data you collect, choose an option under **Ignore time zone and instead use**.
- For **Multiline Processing**, by default only **Boundary Regex** is selected. To make any changes to this setting, select **Detect messages spanning multiple lines** only if the type of data you're collecting is suited to being collected as multi-line log messages.
- If you'd like to filter data being collected, set **Filter** options. See [Filtering data sent from a Source](#) for more information.
- Click **Save** to complete the Source setup.



[Hash and Mask filters](#) can be used to obfuscate proprietary information included in data collected from a Script Source.

When should I set a timeout for my script?

In most cases you won't need to specify a timeout for your script. It's important to note that if a script runs longer than the selected timeout, the next run of the script is canceled, which could lead to a situation where data isn't collected. However, if you're running a long script and it's important to you that your script completes, setting a timeout will cancel the next scheduled run, ensuring that the entire set of data is collected.

Troubleshooting Script Source Issues



To check the status of a Script Source you can look at the Collectors tab of the Sumo Logic Web Application. If you need more information about the reason a Script Source isn't generating log data, you can look at the Collector's logs to help identify what needs to change.

Checking the Collectors Tab to Verify a Source's Operation

The simplest way to confirm that a Source is online, connected to Sumo Logic, and is collecting data is to check the status of the Source in the Collectors tab. Note that if the Source has been configured to run a script once daily and that time has not yet occurred the Source will not yet be online.

To view Source status in the Collectors Tab:

1. In the Sumo Logic Web Application click the Collectors tab.
2. Scroll to the name of the Collector hosting the Script Source.
3. Check to see the icon next to the Source:

	Indicates that the Source is connected to Sumo Logic and that it is generating and collecting log data.
	Indicates that the Source is either not connected to Sumo Logic, or that there is an issue with the script that is preventing data from being collected.

Checking a Collector's Logs to Understand Script Failures

Script-related failures are recorded in the logs generated by the host Collector.

- **Messages related to scripts**

Where a script isn't executable or has failed, or doesn't exist, you'll find messages similar to:

```
Working dir %s does not exist
```

(Where %s is the working directory specified in the Sumo Logic Web Application.)

```
Error in executing script: %s
```

(Where %s is the script.)

- **Messages related to CRON expression errors**

If a CRON expression is invalid a message with information about the issue will be logged.

- **Messages related to timeout errors**

```
Script '%s' failed to start or finish within timeout
```

Advanced Topic: Using CRON Expressions

If you're configuring a Script Source and need to specify a frequency that's different than any existing option, you can specify a CRON expression to collect data at a custom frequency.

Sumo Logic supports the Quartz CRON framework.

To use a CRON Expression in a Script Source:

1. In the Web Application, click the **Collectors** tab, and then click **Add Source** for the Collector you choose.
2. Select **Script** for the Source type.



3. Enter the source's **Name**, **Description** (optional), **Source Host**, and **Source Category**.
4. For **Frequency**, choose **Other (CRON Expression)**, then type the expression.

Frequency* Other (CRON Expression)

Expression 0 0 12 * * ?

☐ Specify a timeout for your command.

Command* /bin/sh

Script * ☐ Type a path to the script to execute.

☒ Type the script to execute.

```
cat myFile.log | grep error | wc -l
```

Working Directory

5. If you'd like to set a timeout for your script, select **Specify a timeout for your command**. If you don't need a timeout, or if you're running a script once daily, we recommend that you leave this option deselected. Learn more in [Configuring a Script Source](#).
6. For **Command**, choose the type of command you're going to use. The options in this menu depend on the type of Collector you're using:

Command*	✓ /bin/sh
	/bin/bash
	/bin/csh
	/usr/bin/ruby
	/usr/bin/perl
	/usr/bin/python

Mac/Linux Command options.

Command*	✓ Windows Script
	Visual Basic Script

Windows Command options.

9. For **Script**, do one of the following:

- Choose **Type a path to the script to execute** if you have the script saved to a file location. For example:

Script * ☒ Type a path to the script to execute.

/Users/username/folder/subFolder/collect-search-interval_2.py

☐ Type the script to execute.

- OR**, choose **Type the script to execute** if you'd like to enter the script directly in the Sumo Logic Web Application. Then type the script in the text box. For example:

Script * ☐ Type a path to the script to execute.

☒ Type the script to execute.

qwinsta.exe

10. For **Working Directory**, you'll only need to enter a path if your script refers to a file indirectly. So, enter the path of the file you'd like to collect if required; otherwise this option can remain blank.

11. Under **Advanced** you'll see options regarding timestamps and time zones:

- Timestamp Parsing.** By default **Extract timestamp information from log file entries** is selected, meaning that Sumo Logic will use the timestamp information from the data you collect. Deselecting this option turns off all timestamp parsing.
- Time Zone.** Select an option under **Use time zone from log file, but if none present use**. Or, if you'd like to override all time zones from data you collect, choose an option under **Ignore time zone and instead use**.

12. For **Multiline Processing**, by default only Boundary Regex is selected. To make any changes to this setting, select **Detect messages spanning multiple lines** only if the type of data you're collecting is suited to being

collected as multi-line log messages.

13. If you'd like to filter data being collected, set **Filter** options. See [Filtering data sent from a Source](#) for more information.
14. Click **Save** to complete the Source setup.

CRON Examples

Use the following examples to structure your CRON expressions.

To run the script at 12:00 PM (noon) every day: 0 0 12 * * ?

To run the script at 11:00 PM every weekday night: 0 23 ? * MON-FRI

To run the script at 10:15 AM every day: 0 15 10 * * ?

To run the script at 10:15 AM every Monday, Tuesday, Wednesday, Thursday and Friday: 0 15 10 ? * MON-FRI

CRON Reference

Cron expressions are comprised of six required fields (seconds, minutes, hours, day of the month, month, day of the week) and one optional field (year) separated by white space:

Field Name	Allowed Values	Allowed Special Characters
Seconds	0-59	- * /
Minutes	0-59	- * /
Hours	0-23	- * /
Day (of month)	1-31	* ? / L W
Month	1-12 or JAN-DEC	- * /
Day (of week)	1-7 or SUN-SAT	- * ? / L #
Year (optional)	empty, 1970-2199	- * /

There are several special characters that are used to specify values:

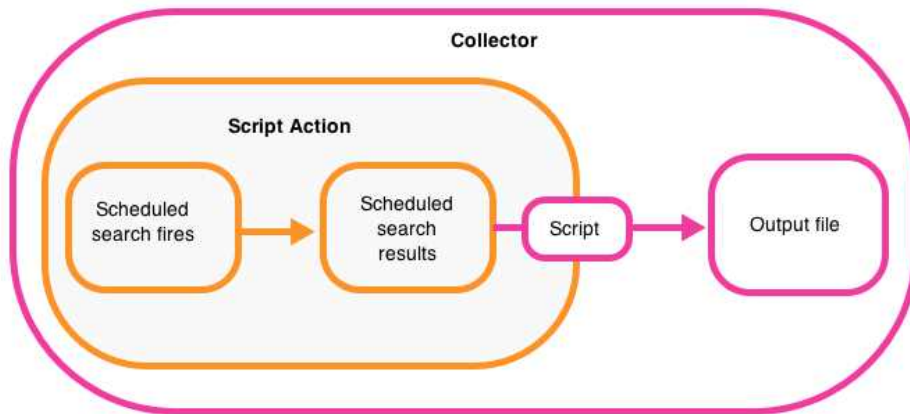
Character	Specifies	Notes
*	All values.	* in the minute field means every minute.
?	No specific value in the day of month and day of week fields.	? specifies a value in one field, but not the other.
-	A range.	10-12 in the hour field means the script will run at 10, 11 and 12 (noon).
,	Additional values.	Typing "MON,WED,FRI" in the day-of-week field means the script will run only on Monday, Wednesday, and Friday.
/	Increments.	0/15 in the seconds field means the seconds 0, 15, 30, and 45. *

		before the '/' is equivalent to specifying 0 is the value to start with. Essentially, for each field in the expression, there is a set of numbers that can be turned on or off. For seconds and minutes, the numbers range from 0 to 59.
#	Day of a month.	6#3 in the day of week field means the third Friday (day 6 is Friday; #3 is the 3rd Friday in the month). If you specify, say #5, and there isn't a 5th occurrence of the given day, the CRON job won't fire. If # is used, there can only be one expression in the day of week field.
L	The last day of a month or week.	L means the last day of the month. If used in the day of week field by itself, it means 7 or SAT. If used in the day of week field after another value, it means the last [day] of the month; for example 6L means the last Friday of the month. You can also specify an offset from the last day of the month; L-3 means the third to last day of the month. Make sure not to use L to specify lists or ranges of values.
W	The weekday (Mon-Fri) nearest the specified day.	Specifying 15W means the CRON job will fire on the nearest weekday to the 15th of the month. If the 15th is a Saturday, the trigger fires on Friday the 14th. If the 15th is a Sunday, the trigger fires on Monday the 16th. W can only be specified when the day of month is a single day (not a range or list of days).

Configuring a Script Action

Unlike all other Source varieties, a Script Action receives data uploads triggered by a scheduled search. The script you create defines how data is consumed; for example, you could fire SNMP traps based on the result of the search.

After setting up an Script Action, the next step is to construct a scheduled search. Each time the search query executes, the Collector runs the script configured in the Script Action.



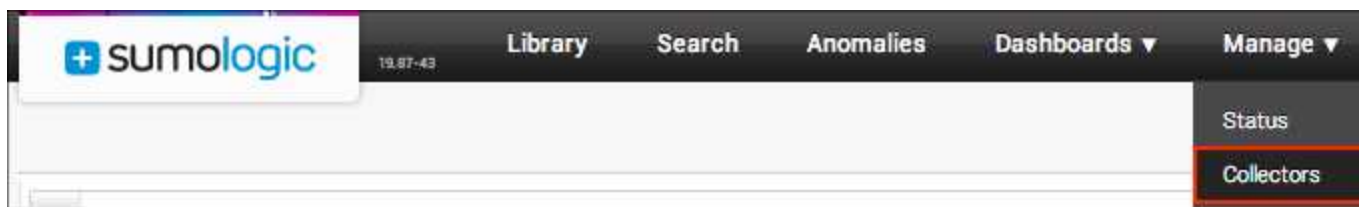
Step 1. Set up the Script Action.

Script Actions can be added to any Collector running the latest version of Sumo Logic Collector software.

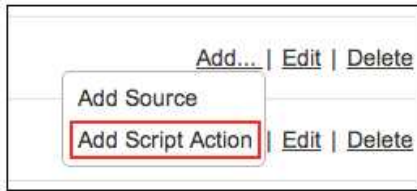
When creating the script, make sure the script parses the search results in a manner that best suits your organization, keeping in mind that the file is in JSON format. The script configured in the Script Action is permitted to call other external scripts.

To configure a Script Action:

1. In the Web Application, select **Manage > Collectors**.



2. Find the name of the installed Collector to which you'd like to add a Source. Click **Add...** then choose **Add Script Action** from the pop-up menu.



3. Enter a **Name** to display for the Script Action. Description is optional.
4. If you'd like to specify a timeout for your script, select **Specify a timeout for your command**. Setting a timeout ensures that a script is killed, making sure that resources aren't fully consumed. If you choose to set a timeout, make sure to select a generous amount of time to make sure that the script has enough time to finish running.
5. For **Command**, choose the type of command you're going to use.
6. In the **Script** text box, type the script's path. The script itself cannot be typed in the Script text box.
7. For **Working Directory**, specify a directory if you need your script action to execute in a different directory than the Collector's install directory.
8. Click **Save**.

Step 2. Set up a Scheduled Search.

Once the Script Action has been added to your Sumo Logic account, it's time to create a scheduled search. Note that the Search name will appear in the output file, along with the query.

The first time the scheduled search executes, output files will begin to be generated.

To schedule a search:

1. In the Search page, type the search you'd like to use with the Script Action, then click **Save As** below the search field.
2. For **Search name**, enter a name for the search. If you'd like, type an optional description to help you identify this search.
3. Choose a **Time Range** option that will be the default range when you run the saved search.
4. Choose an option from the **Run frequency** menu.
5. Choose a **Time Range** option for the frequency the saved search should be run.
5. For **Alert Condition**, select **Notify me every time upon search completion** to make sure that alerts are sent to the Script Action.
6. For **Alert Type**, select **Script Action**.
7. For **Script Action**, select the name of the Script Action (displayed with its Collector's name) from the pop up menu.
8. Click **Save**.

Save Search

Search name*
Message Piles

Description

Search*
Publishing message piles | parse "messages: '*'" as numOfMsg

Timerange
Last 15 Minutes

Folder
PERSONAL
Log Analysis Quick...
Nginx

Run frequency
Every 15 Minutes At:

Timerange*
Last 15 Minutes

Alert condition
☒ Send notification every time upon search completion
☐ Send notification only if the condition below is satisfied:
 Number of results (Count to >)

Alert Type
Script Action

Script Action*
Script

Cancel | Save

About the search results file

The Sumo Logic file is the result of a scheduled search written in JSON format. It includes the results of the scheduled search, as well as information about the time range of the search. By default, the files are stored in the Collector installation directory. Every three hours the files are purged.

A maximum of 10,000 messages are included in the file. Each message in the search results is marked with the Collector's metadata and a time stamp. At the end of each file you'll find information about the scheduled search:

```

{
  "Collector": "Gangnam",
  "Host": "NorCal",
  "Message": "2013-01-22 10:21:13,406 [Thread-8] INFO com.gangnam.collector.CommonsHTTPSender - Publishing message piles: '1', messages: '1', bytes: '409', encoded: '395',",
  "Name": "/Users/mtn_view/Development/sumo/collector/logs/collector.log",
  "Time": "1358878873406",
  "numofmsg": "1"
},
{
  "Category": "collector",
  "Collector": "Gangnam",
  "Host": "NorCal",
  "Message": "2013-01-22 10:21:11,408 [Thread-8] INFO com.gangnam.collector.CommonsHTTPSender- Publishing message piles: '1', messages: '1', bytes: '409', encoded: '395',",
  "Name": "/Users/mtn_view/Development/sumo/collector/logs/collector.log",
  "Time": "1358878871408",
  "numofmsg": "1"
}
],
"resultsEndAt": 1358878980000,
"resultsStartAt": 1358878880000,
"runAs": "Miranda",
"searchName": "MessagePiles",
"searchQuery": "Publishing message piles I parse \"messages: '*'\" as numOfMsg",
"searchUrl": "https://index.html#section/search/DtuMAqRfDpeeP1hS1Z6fyH7sS73tQUjJZEhEvVfMa3pIW1aFiWw2xAVB2Z9dJzfBDMziMwJNSpgcaexz",
"totalCount": "1596",
"sessionId": "CCFC8950C254C72A"
}

```

- A. End of scheduled search (Unix timestamp)
- B. Beginning time of scheduled search.
- C. User account.
- D. Name of the scheduled search (reflects the name saved with the search; can be modified)
- E. Query saved as the scheduled search.
- F. Click the URL to view the results of the search in a web page.
- G. Number of messages.

Adding an Amazon S3 Source

After setting up a Hosted Collector, you'll add an Amazon S3 Source. Before adding the Source, make sure that your AWS account is set up properly. Learn more in [Granting access to an S3 bucket](#).

Configuring an Amazon S3 Source

After setting up a Hosted Collector, you'll need to configure an S3 Source to begin collecting data from an S3 bucket.

Note that one S3 Source can collect data from just a single S3 bucket. However, you can configure multiple S3 Sources to collect from one S3 bucket. For example, you could use one S3 Source to collect one particular data type, and then configure another S3 Source to collect another data type.

Text files and compressed text (gzip) files can be uploaded to an S3 Source.

Before you begin

In order to configure an S3 Source you first need to make sure that you've given Sumo Logic privileges to access your organization's S3 bucket. This access is granted through **Amazon Web Service Identity and Access Management (IAM)**. See [Granting Access to an S3 bucket](#) for more information and instructions.

About Path Expressions in S3 Sources

Sumo Logic uses the **Path Expression** to identify which objects should be uploaded from an S3 bucket to the service. You can type the exact name of just one specific object if that's what you'd like to collect. Alternately, you can type a wildcard (*) in the string to collect multiple objects. Note that using two wildcards together (**) is not supported.

For example:

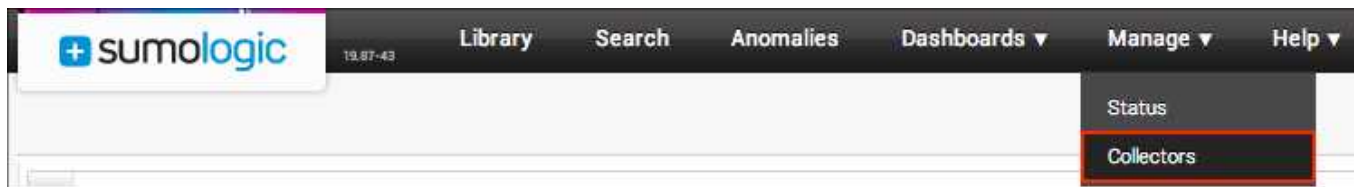
- Typing `postgresql-2013-03-01_000000.log.gz` would *only* fetch the object with that exact name.
- Typing `postgresql*.log*` would fetch objects such as `postgresql-2013-03-01_000000.log.gz`, `postgresql-2013-03-01_000000.log`, `postgresql.log`, and so on.
- Typing `*postgresql*.log*` would fetch objects listed above, as well as others like `/old/2011/postgresql.log`, `/old/postgresql.log`, etc.



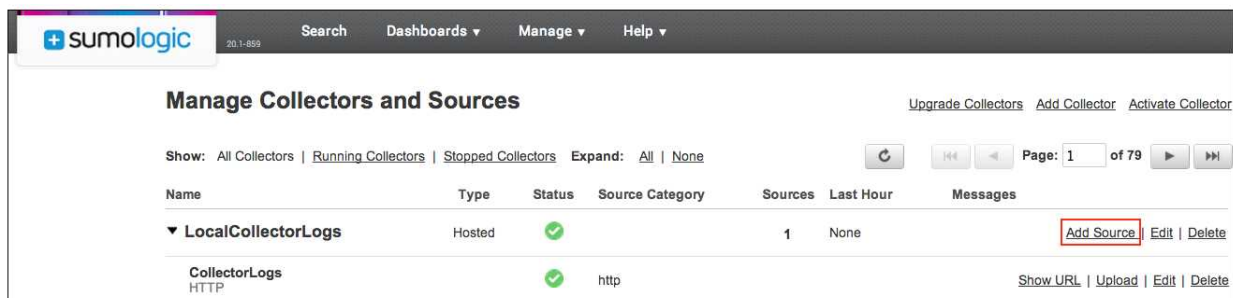
If you'd like to collect from a "folder" within an S3 bucket, the path expression should not start with a leading forward slash. For example, `/name/*` won't find the files; instead use `"name/*"`.

Configuring an S3 Source

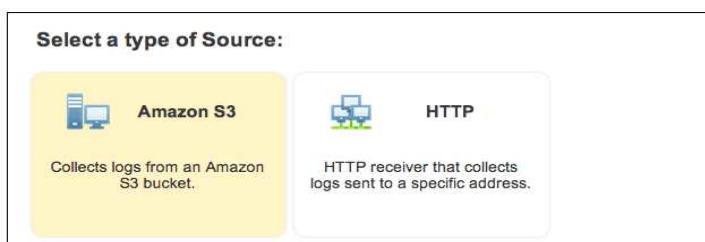
1. In the Sumo Logic Web Application select **Manage > Collectors**.



- In the Collectors page, click **Add Source** next to a **Hosted Collector**.



- Select **Amazon S3**.



- Enter a **Name** to display for the new Source. Description is optional.
- For **Bucket Name**, type the exact name of your organization's S3 bucket. Be sure to double-check the name as it appears in AWS, for example:



- For **Path Expression**, type the string that matches the S3 objects you'd like to collect. A wildcard (*) can be used in this string. [See About Path Expressions](#) for more information.
- For **Source Category**, enter any string to tag the output collected from this Source. (Category metadata is stored in a searchable field called `_sourceCategory`.)
- For **Key ID**, type the [AWS Access Key ID](#) number granted to Sumo Logic. (See [Granting access to an S3 bucket](#) for more information.)

9. For **Secret Key**, type the [AWS Secret Access Key](#) Sumo Logic should use to access the S3 bucket. (See [Granting access to an S3 bucket](#) for more information.)
10. For **Scan Interval**, use the default of 5 minutes. Alternately, type the frequency Sumo Logic will scan your S3 bucket for new data. To learn more about Scan Interval considerations, see [About setting the S3 Scan Interval](#).
11. Set any of the following under **Advanced**:

Enable Timestamp Parsing. This option is selected by default. If it's deselected, no timestamp information is parsed at all.

Time Zone. There are two options for Time Zone. You can use the time zone present in your log files, and then choose an option in case time zone information is missing from a log message. Or, you can have Sumo Logic completely disregard any time zone information present in logs by forcing a time zone. It's very important to have the proper time zone set, no matter which option you choose. If the time zone of logs can't be determined, Sumo Logic assigns logs UTC; if the rest of your logs are from another time zone your search results will be affected.

Timestamp Format. By default, Sumo Logic will automatically detect the timestamp format of your logs. However, you can manually specify a timestamp format for a Source. See [Specifying a Timestamp Format](#) for more information.

Multiline Processing. Deselect **Multiline Processing** to avoid unnecessary processing if you are collecting single-message-per-line files (for example, Linux system.log). If you're working with multi-line messages (for example, log4J or exception stack traces) then select this option.

Infer Boundaries. Enable when you want Sumo Logic to automatically attempt to determine which lines belong to the same message.



If you deselect the Infer Boundaries option, you will need to enter a regular expression in the Boundary Regex field to use for detecting the entire first line of multi-line messages.

Boundary Regex. You can specify the boundary between messages using a regular expression. Enter a regular expression for the full first line of every multi-line message in your log files. For an example, see [Boundary Regex](#).

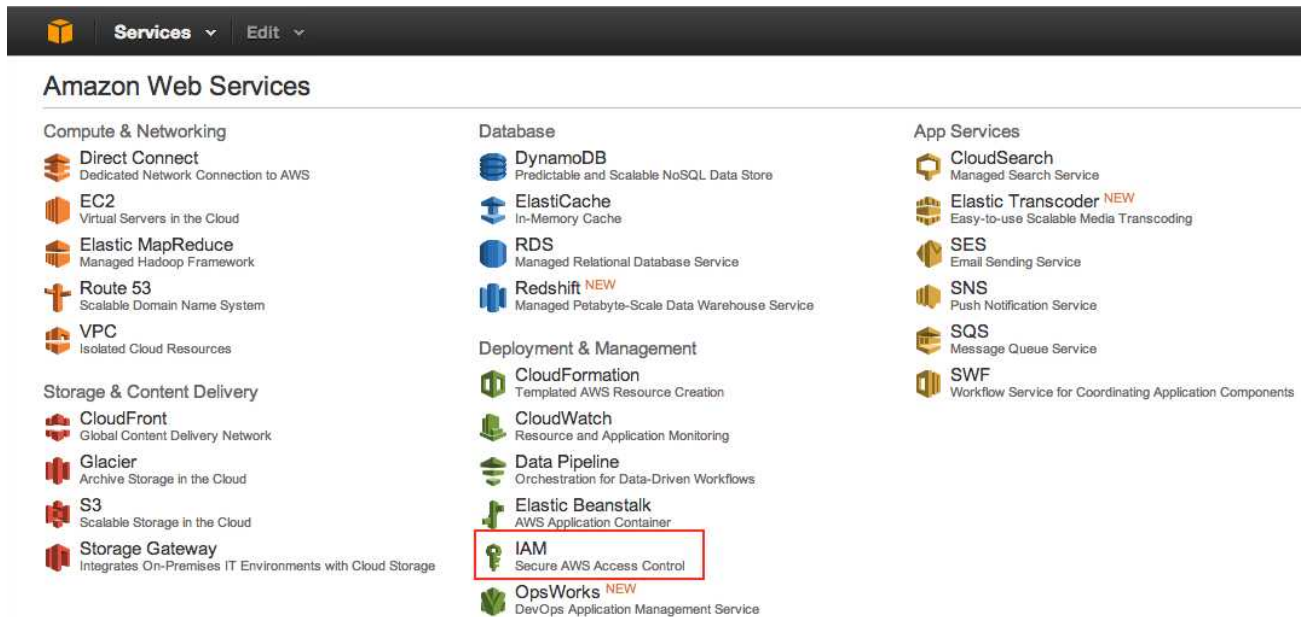
12. [Create any filters](#) you'd like for the S3 Source.
13. When you are finished configuring the Source click **Save**.

Granting Access to an S3 bucket

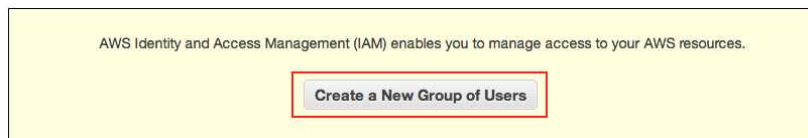
Before configuring an S3 Source, you'll need to grant Sumo Logic permissions to get objects and object versions, and list object and object versions in your organization's bucket. First you'll need to make sure that your AWS account includes IAM. You can enable IAM through the AWS control panel.

Granting S3 permissions

1. In your AWS account Services page, click **IAM**.



2. Click **Create a New Group of Users**.



3. Create a group named SumoLogic, then click **Continue**.



4. For Permissions, choose **Custom Policy**, then click **Select**.

Create a New Group of Users Cancel X

Group Name Permissions Users Review

Set Permissions

Select a policy template, generate a policy, or create a custom policy. A policy is a document that formally states one or more permissions. You can edit the policy on the following screen, or at a later time using the user, group, or role detail pages.

☐ Select Policy Template

☐ Policy Generator

☒ Custom Policy

Use the policy editor to customize your own set of permissions. Select

☐ No Permissions

- For Policy Name, you may want to use "allow-s3-read" or something similar, so your organization is aware of why this policy was created. Then, enter the JSON parameters you'd like to use for the policy (see this [JSON example](#) to copy and paste a recommended policy). Click **Continue**.

Create a New Group of Users Cancel X

Group Name Permissions Users Review

Set Permissions

You can customize permissions by editing the following policy document. For more information about the access policy language, see [Overview of Policies](#) in Using IAM.

Policy Name

allow-s3-read

Policy Document

```
{
  "Statement": [
    {
      "Sid": "Stmt1366678552407",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:ListBucketVersions",
        "s3:ListBucket"
      ]
    }
  ]
}
```

[Back](#) Continue

- Next, create a user (we named our sumologic-bucket-scanner) making sure to generate a key. Then click **Continue**.

Create a New Group of Users Cancel

Group Name Permissions **Users** Review

Users below will be added to your sfg group.

Create New Users Add Existing Users

Enter User Names:

1.

2.

3.

4.

5.

Maximum 128 characters each

☒ **Generate an access key for each User**

Users need access keys to make secure REST or Query protocol requests to AWS service APIs.

For Users who need access to the AWS Management Console, create a password in the Users panel after completing this wizard.

Back Continue

7. Click **Continue**. The group is created.
8. Click **Show User Security Credentials** to view the **Access Key ID** and **Secret Access Key** for this user. You'll provide it to Sumo Logic. You can also choose to download a .csv file with this information by clicking **Download Credentials**.

Create a New Group of Users Cancel

✓ Your 1 User(s) have been created successfully.

This is the last time these User security credentials will be available for download.

You can manage and recreate these credentials any time.

▼ Hide User Security Credentials

sumologic-bucket-scanner

Access Key Id: AKIAJQQCQ5UGI

Secret Access Key: +p8bNeH7nbPgTCZ7ZZr5P8xA8t8FifR

Download Credentials Close Window

Policy JSON

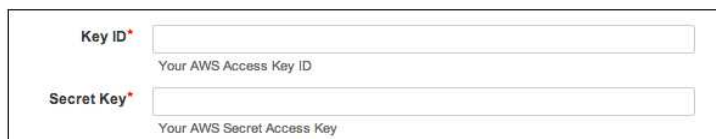
We recommend using the following JSON to create a policy:

```
{
  "Statement": [
    {
      "Sid": "Stmt1366678552407",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:ListBucketVersions",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [ "arn:aws:s3:::your_bucketname/*", "arn:aws:s3:::your_
bucketname" ]
    }
  ]
}
```

Make sure to enter the actual name of your S3 bucket to the Resource line of JSON.

Managing Access Keys

In addition, while configuring an S3 Source, you'll need to provide **Key ID** and **Secret Key** credentials (tokens) to Sumo Logic. Security, token, and access settings are handled through **Amazon Web Service Identity and Access Management (IAM)**.



The image shows a form with two input fields. The first field is labeled 'Key ID*' and has a placeholder text 'Your AWS Access Key ID'. The second field is labeled 'Secret Key*' and has a placeholder text 'Your AWS Secret Access Key'.

Important: If your organization does not yet have IAM in your AWS account, you must add this option before configuring an S3 Source. Otherwise Sumo Logic won't have appropriate permissions to access your data.

For instructions on using IAM , please see [AWS Identity and Access Management \(IAM\)](#) to learn about the options available to your organization.

About setting the S3 Scan Interval

When configuring an S3 Source, you'll set the **Scan Interval**, which defines the waiting time between scans of the objects in your S3 bucket. It's important to set an interval that is long enough to allow new files to be uploaded, but is not too short that scans are performed without any new files being available to upload.

Setting a Scan Interval that's too *short* could cause additional charges to your AWS account. When Sumo Logic scans the contents of a bucket for new files, it will perform a number of listings, which may increase the number of objects in the bucket. Sumo Logic can't determine if the data in your S3 bucket has changed without listing each object in every Scan Interval.

In addition, be aware that uploading data to Sumo Logic can incur data transfer charges from AWS as well. For your information, you can view current pricing for list and data transferring [here](#). To get an idea of what your charges could be, we recommend using the [Simple Monthly Calculator](#).

Setting a Scan Interval that's too *long* can cause a delay in new files being uploaded in a timely manner. If no new files are found in a scan, the Scan Interval is automatically doubled, up to a maximum of 24 hours. For example, let's say your Scan Interval is set to 10 minutes. After a scan is completed with no new files identified, the Scan Interval goes to 20 minutes. Likewise, if no new files are found in 20 minutes, the Scan Interval changes to 40 minutes. This continues up until the interval is set 24 hours, which means that uploading a new file could be delayed up to 24 hours.

Configuring an HTTP Source

An HTTP Source is an endpoint for receiving a file (or a batch of files) uploaded via a unique URL generated for the Source. The URL securely encodes the Collector and Source information. You can add as many HTTP Sources as you'd like to a single Hosted Collector.

With an HTTP Source you can upload logs from data sources where you cannot install a Local Collector. For example, you can export data from a platform as a service (PaaS) or infrastructure as a service (IaaS) provider, allowing you to gain visibility from, say your billing system service provider, leveraging the same Sumo Logic tools your organization already uses. Please check with your IaaS or PaaS providers for information regarding using their APIs to forward log data into Sumo Logic's HTTP endpoint.

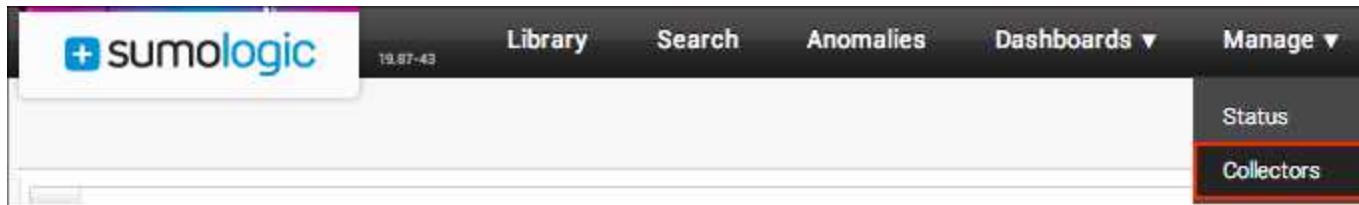
The generated URL is a long string of letters and numbers. You can generate a new URL at any time. For more information see [Generating a new URL](#).

Adding an HTTP Source

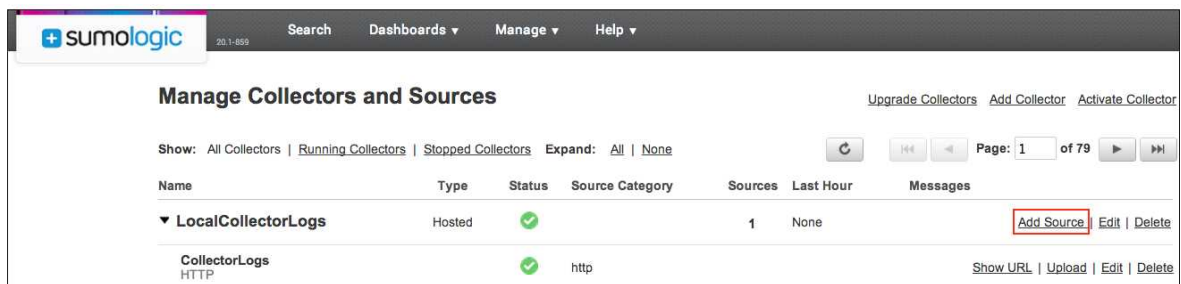
When you set up an HTTP Source, a unique URL is assigned to that Source. When you upload a file using that URL, it's associated with the Source, and metadata is tagged to the file.

To configure an HTTP Source:

1. In the Sumo Logic Web Application select **Manage > Collectors**.



2. In the Collectors page, click **Add Source** next to a **Hosted Collector**.



3. Select **HTTP**.

Select a type of Source:


Amazon S3
Collects logs from an Amazon S3 bucket.


HTTP
HTTP receiver that collects logs sent to a specific address.

4. Enter a **Name** to display for this Source in the Sumo Logic Web Application. Description is optional.
5. (Optional) For **Source Host** and **Source Category**, enter any string to tag the output collected from this Source. (Category metadata is stored in a searchable field called `_sourceCategory`.)
6. Set any of the following under **Advanced**:

Enable Timestamp Parsing. This option is selected by default. If it's deselected, no timestamp information is parsed at all.

Time Zone. There are two options for Time Zone. You can use the time zone present in your log files, and then choose an option in case time zone information is missing from a log message. Or, you can have Sumo Logic completely disregard any time zone information present in logs by forcing a time zone. It's very important to have the proper time zone set, no matter which option you choose. If the time zone of logs can't be determined, Sumo Logic assigns logs UTC; if the rest of your logs are from another time zone your search results will be affected.

Timestamp Format. By default, Sumo Logic will automatically detect the timestamp format of your logs. However, you can manually specify a timestamp format for a Source. See [Specifying a Timestamp Format](#) for more information.

Multiline Processing. Deselect **Multiline Processing** to avoid unnecessary processing if you are collecting single-message-per-line files (for example, Linux system.log). If you're working with multi-line messages (for example, log4J or exception stack traces) then select this option.

Infer Boundaries. Enable when you want Sumo Logic to automatically attempt to determine which lines belong to the same message.



If you deselect the Infer Boundaries option, you will need to enter a regular expression in the Boundary Regex field to use for detecting the entire first line of multi-line messages.

Boundary Regex. You can specify the boundary between messages using a regular expression. Enter a regular expression for the full first line of every multi-line message in your log files. For an example, see [Boundary Regex](#).

Enable One Message Per Request. Leave this choice deselected if you would like Select this option if you'll be sending a single message with each HTTP request.

7. [Create any filters](#) you'd like for the new Source.
8. When you are finished configuring the Source click **Save**.
9. When the URL associated with the Source is displayed, copy the URL so you can use it to [send files](#).

HTTP Source Address

Use the following address to send data to the Collector.

`https://sumologic.net/receiver/v1/http/ZaVnC4dhaV`

OK

10. Choose the method you'll use to upload files to the Source.



If you need to access the Source's URL again, just click Show URL in the Collector tab of the Web Application:

Manage Collectors and Sources

[Upgrade Collectors](#) [Add Collector](#) [Activate Collector](#)

Show: [All Collectors](#) | [Running Collectors](#) | [Stopped Collectors](#) Expand: [All](#) | [None](#)

Name	Type	Status	Source Category	Sources	Last Hour	Messages
▼ 0-CHR-TEST-HTTP	Hosted	✓		2	None	Add Source Edit Delete
FIRST HTTP		✓	http			Show URL Edit Delete

Uploading data to an HTTP Source

Once an HTTP Source has been added to a Hosted Collector you can begin uploading data. All files are uploaded over HTTPS, meaning that your data is secure.

In addition to plain text, HTTP Sources support **gzip** and deflate compressed data. When uploading plain text files, there are no prerequisites. However, if you upload gzip or deflate compressed data, you must specify "gzip" or "deflate," respectively, in the **Content-Encoding** header. For more information, see [Passing parameters in HTTP headers](#).

To upload data to an HTTP Source:

Choose one of the three methods for uploading data to the unique URL assigned to the HTTP Source:

- **Script.** Construct a Java script that points to the URL.
- **cURL.** POST to the URL associated with the Source. For example:
`curl -X POST -T your-file-here https://sumologic.net/v1/http/your_end_point_here`
- **GET.** In your command, everything after the question mark (?) is considered to be the data that needs to be ingested. For example, if your command is:
`curl -v https://stage-00000000472.collection.sumologic.net/receiver/v1/http/XXXXXX?thisismypayload`
the data named **thisismypayload** will be uploaded.

Passing parameters in HTTP headers

There are two parameters that can be passed as a header in the HTTP request called.

- **Uploading gzip or compressed data.** In the header, specify **Content-Encoding: gzip** or **Content-Encoding: deflate**.
- **Adding a name.** To add a name to an uploaded message, file, or batch of files, specify **X-Sumo-Name: name**.

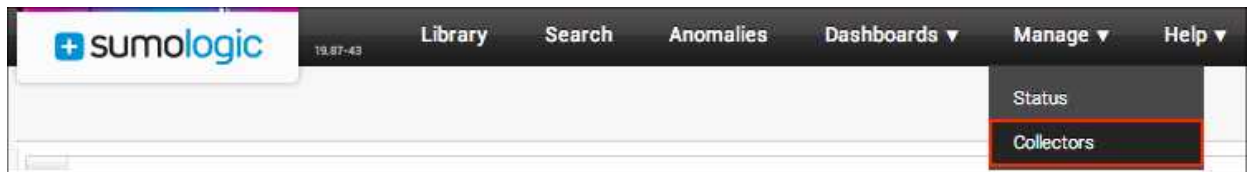
Generating a new URL

You can generate a new URL for an HTTP Source at any time. Note that generating a new URL completely invalidates the old URL.

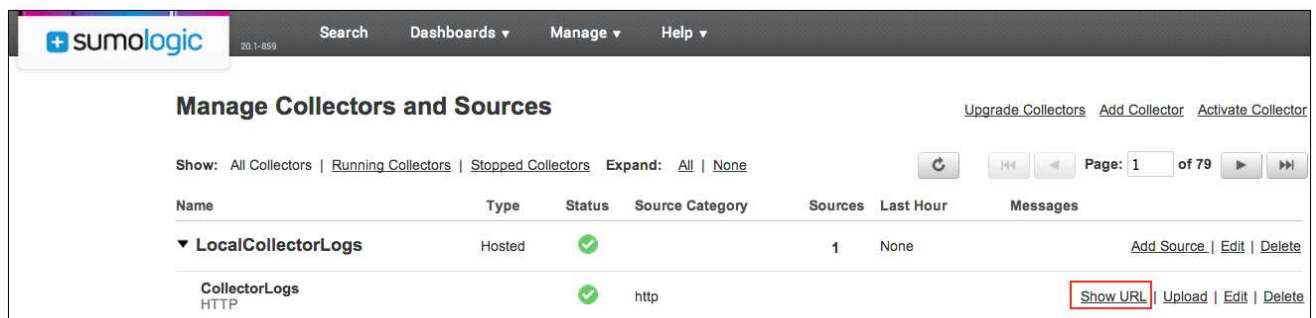
Note that if you see a 401 (failed to authenticate) error message right after generating a new URL, wait a few minutes, then try the new URL again.

To generate a new URL:

1. In the Sumo Logic Web Application select **Manage > Collectors**.



2. In the Collectors page of the Sumo Logic Web Application click **Show URL** next to the HTTP Source.



3. In the HTTP Source Address dialog box, click **Generate**.

HTTP Source Address

Use the following address to send data to the Collector.

`https://sumologic.net/receiver/v1/http/ZaVnC4dhaV:`

Click the 'Generate' button to generate a new URL and invalidate any previous ones

Generate **OK**

4. When asked to confirm the generation, click **OK**.
5. In the HTTP Source Address dialog box, the new URL is displayed. Copy and paste the URL, then click **OK**.

HTTP Source Address

Use the following address to send data to the Collector.

`https://sumologic.net/receiver/v1/http/MrkLIaM8MG`

Click the 'Generate' button to generate a new URL and invalidate any previous ones

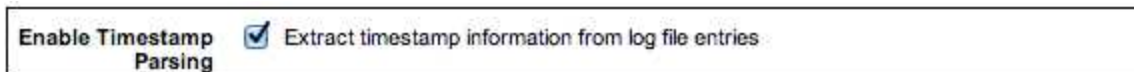
OK

Setting Source timestamp options

When configuring a Source you can choose to use the Sumo Logic default timestamp parsing settings, or you can specify a custom format for Sumo Logic to parse timestamps in your log messages. In addition, you can choose how the time zone in your log data is handled.

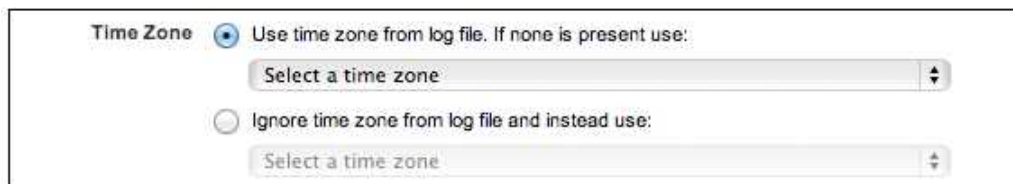
Source timestamp and time zone options include:

Enable Timestamp Parsing. This option is selected by default. If it's deselected, no timestamp information is parsed at all. Instead, Sumo Logic stamps logs with the time at which the messages are processed.



Enable Timestamp Parsing ☒ Extract timestamp information from log file entries

Time Zone. There are two options for Time Zone. You can use the time zone present in your log files, and then choose an option in case time zone information is missing from a log message. Or, you can have Sumo Logic completely disregard any time zone information present in logs by forcing a time zone. It's very important to have the proper time zone set, no matter which option you choose. If the time zone of logs can't be determined, Sumo Logic stamps them with the Pacific Time Zone; if the rest of your logs are from another time zone your search results will be affected.



Time Zone ☒ Use time zone from log file. If none is present use:
Select a time zone
☐ Ignore time zone from log file and instead use:
Select a time zone

Timestamp Format. By default, Sumo Logic will automatically detect the timestamp format of your logs. However, in cases where timestamps are not parsing correctly, you can manually specify a timestamp format for a Source. See [Specifying a Timestamp Format](#) for more information



Timestamp Format ☐ Automatically detect the format ☒ Specify a format
Format
MM/dd/yyyy hh:mm:ss a
Enter a format to use for parsing. For example 'yyyy-MM-dd HH:mm:ss'
Log Sample
9/29/2012 10:23:15 AM Test
Enter a date time string from your logs, then click Test to validate it.
For example '2012-07-04 14:33:22'

Specifying a timestamp format

In the majority of cases Sumo Logic automatically parses timestamps without any issues, but if you're seeing timestamp parsing issues you can manually specify the parse format. The steps are the same if you're configuring

a new Source or if you're editing the timestamp information for an existing Source.

To manually specify a timestamp format for a Source:

1. Do one of the following:
 - If you're configuring a new Source, continue to step 2.
 - To edit the timestamp settings for an existing Source, click the Collectors tab of the Sumo Logic Web Application. Then click **Edit** to the right of the Source name.
2. Click **Advanced**.
3. For **Timestamp Format**, select **Specify a format**.
4. In the **Format** text box, type the timestamp format that Sumo Logic should use to parse timestamps in your logs.
5. To verify that the format is supported, and that it can be properly parsed, copy and paste a timestamp string from your log file in the **Log Sample** text box. Then click **Test**.
Sumo Logic looks at the format to make sure it's supported, then tests the sample timestamp against the format to make sure it's valid, and that it can be parsed. You'll be notified if the test is successful or if there are any issues that need to be addressed before continuing.
6. Click **Save**.

Supported Timestamp Conventions

As long as the following conventions are followed, Sumo Logic can parse timestamps from log messages:

Letter	Date or Time Component	Example
y	Year	2012; 12
M	Month (in a year)	October; Oct, 10
D	Day (in a year)	174
d	Day (in a month)	16
a	AM/PM	PM
H	Hour (in a day; 0-23)	2
k	Hour (in a day; 1-24)	24
K	Hour (in AM/PM; 0-11)	0
h	Hour (in AM/PM; (1-12)	12
m	Minute (in an hour)	40
s	Second (in a minute)	35
S	Millisecond	875
z	Time zone (General time zone)	Pacific Standard; PST; GMT-09:00
Z	Time zone (RFC 822 time zone)	-0900

Are Unix timestamps supported?

Yes, Unix timestamps are supported. These timestamps must begin with "[" and end with a "," with 10 digits in between.

Timestamp examples

In addition to custom formats using the components listed above, any of the following timestamp formats can be parsed by Sumo Logic:

Timestamp Format	Example
dd/MMM/yyyy:HH:mm:ss ZZZZ	19/Apr/2010:06:36:15 -0700
dd/MMM/yyyy HH:mm:ss	09/Mar/2004 22:02:40 08691
MMM dd, yyyy hh:mm:ss a	Dec 2, 2010 2:39:58 AM
MMM dd yyyy HH:mm:ss	Jun 09 2011 15:28:14
MMM dd HH:mm:ss yyyy	Apr 20 00:00:35 2010
MMM dd HH:mm:ss ZZZZ	Sep 28 19:00:00 +0000
MMM dd HH:mm:ss	Mar 16 08:12:04
yyyy-MM-dd HH:mm:ss,SSS ZZZZ	2011-02-11 16:47:35,985 +0000
yyyy-MM-dd HH:mm:ss ZZZZ	2011-08-19 12:17:55 -0400
yyyy-MM-dd HH:mm:ssZZZZ	2011-08-19 12:17:55-0400
yyyy-MM-dd HH:mm:ss,SSS	2010-06-26 02:31:29,573
yyyy-MM-dd HH:mm:ss	2010-04-19 12:00:17
yyyy/MM/dd HH:mm:ss	2006/01/22 04:11:05
yy-MM-dd HH:mm:ss,SSS ZZZZ	11-02-11 16:47:35,985 +0000
yy-MM-dd HH:mm:ss,SSS	10-06-26 02:31:29,573
yy-MM-dd HH:mm:ss	10-04-19 12:00:17
yy/MM/dd HH:mm:ss	06/01/22 04:11:05
HH:mm:ss,SSS	11:42:35,173
MM/dd/yyyy hh:mm:ss a:SSS	8/5/2011 3:31:18 AM:234
MM/dd/yyyy hh:mm:ss a	9/28/2011 2:23:15 PM

Using JSON to Configure Sources

The JSON files used to configure Sources follow the syntax used in the [Collector Management API](#). Samples are provided for each Source type below.



JSON parameters are only used the first time a Collector is set up. Any later modifications made to JSON will not be picked up by the Collector. To make changes to Collector settings after the Collector has been configured, you can use the Collector Management API or the Sumo Logic Web Application.

Non-configurable parameters

The following parameters should not be manually configured. They are automatically configured by the Sumo Logic Service.

- id
- alive
- status

Time zone format

Sumo Logic uses the TZ database time zone format. For example, PST (Pacific Standard Time) is expressed as **America/Los_Angeles**. You can find a list of TZ environment variables [here](#).

Generic parameters

The following parameters are used for all Source types.

Parameter	Type	Required?	Description
name	String	Yes	Type the name of the Source, like "SourceName",.
description	String	No	Type a description of the Source, like "SourceDescription",.
category	String	No	Type the description of the category of the Source. For example, you could type "auth_logs", r "mail",.
hostName	String	No	Type the host name of the Source, like "SourceHost",.
timeZone	String	No	Type the time zone you'd like the Source to use in TZ database format. For example, America/Los_Angeles .
sourceType	String	Yes	Type the correct type of Source (see specific Source types below for more information).
automaticDateParsing	Boolean	No	Determines if timestamp information is parsed or not. Type true to enable automatic parsing of dates (the default setting); type false to disable. If disabled, no timestamp information is parsed at all.
multilineProcessingEnabled	Boolean	No	Type true to enable; type false to disable. The default setting is true. Consider setting to false avoid unnecessary processing if you are collecting single-message-per-line files

			(for example, Linux system.log). If you're working with multi-line messages (for example, log4J or exception stack traces) then keep this setting enabled.
useAutolineMatching	Boolean	No	Type true to enable if you'd like message boundaries to be inferred automatically; type false to prevent message boundaries from being automatically inferred (equivalent to the Infer Boundaries option in the UI). The default setting is true .
manualPrefixRegexp	String	No	A type a regular expression for the prefix of a message. Alternately, type "", to leave this empty.
forceTimeZone	Boolean	No	Type true to force the Source to use a specific time zone, otherwise type false to use the time zone found in the logs. The default setting is false .
defaultDateFormat	String	No	Type the default format for dates used in your logs. For more information about timestamp options, see Supported Time Stamp Conventions .
cutoffTimestamp	Long	No	This option allows you to specify the timestamp of the oldest logs you'd like to collect. Type the milliseconds since epoch of the oldest log.
cutoffRelativeTime	String	No	If you'd like to provide a relative cutoff of the logs you'd like to collect, type "-1h", to collect data that's been modified less than one hour old, "-1d", to collect data that's been modified less than one day old, or even "-1w", to collect data that's been modified less than one week old. Note that the parameter does not use the message timestamp, but instead uses the last modified timestamp.
filters	Boolean	No	If you'd like to add a filter to the Source, type the name of the filter ("Exclude", "Include", "Mask", or "Hash"). See the Collector Management API Read Me for more information.

Local File Source

Parameter	Type	Required?	Description
pathExpression	String	Yes	A valid path expression of the file to collect.
blacklist	String array	No	Comma-separated list of valid path expressions from which logs will not be collected. For example, "blacklist": ["/var/log/**/*bak", "/var/oldlog/*.log"]
sourceType	String	Yes	Type "LocalFile".

name	String	Yes	Name of the Source.
description	String	No	Type a description of the Source, like "SourceDescription",.
category	String	No	Type the description of the category of the Source. For example, you could type "auth_logs", r "mail",.
hostName	String	No	Type the host name of the Source, like "SourceHost",.
timeZone	String	No	Type the time zone you'd like the Source to use in TZ database format. For example, America/Los_Angeles .
automaticDateParsing	Boolean	No	Determines if timestamp information is parsed or not. Type true to enable automatic parsing of dates (the default setting); type false to disable. If disabled, no timestamp information is parsed at all.
multilineProcessingEnabled	Boolean	No	Type true to enable; type false to disable. The default setting is true. Consider setting to false avoid unnecessary processing if you are collecting single-message-per-line files (for example, Linux system.log). If you're working with multi-line messages (for example, log4J or exception stack traces) then keep this setting enabled.
useAutolineMatching	Boolean	No	Type true to enable if you'd like message boundaries to be inferred automatically; type false to prevent message boundaries from being automatically inferred (equivalent to the Infer Boundaries option in the UI. The default setting is true.
manualPrefixRegex	String	No	A type a regular expression for the prefix of a message. Alternately, type "", to leave this empty.
forceTimeZone	Boolean	No	Type true to force the Source to use a specific time zone, otherwise type false to use the time zone found in the logs. The default setting is false .
defaultDateFormat	String	No	Type the default format for dates used in your logs. For more information about timestamp options, see Supported Time Stamp Conventions .
cutoffTimestamp	Long	No	This option allows you to specify the timestamp of the oldest logs you'd like to collect. Type the milliseconds since epoch of the oldest log.
cutoffRelativeTime	String	No	If you'd like to provide a relative cutoff of the logs you'd like to collect, type "-1h", to collect data that's been modified less than one hour old, "-1d", to collect data that's been modified less than one day old, or even "-1w", to collect data that's been modified less than one week old. Note that the parameter does not use the message timestamp, but instead uses the last modified timestamp.
filters	Boolean	No	If you'd like to add a filter to the Source, type the name of the filter ("Exclude", "Include", "Mask", or "Hash"). See the Collector Management API Read Me for more information.

Local File Source JSON example:

```
{
  "api.version": "v1",
  "sources": [
    {
      "sourceType" : "LocalFile",
      "name" : "Example1",
      "pathExpression" : "/var/logs/maillog",
      "blacklist":["/var/log/*log1.log"],
      "/var/log/log2.log"],
      "category": "mail",
      "hostName": "sampleSource",
      "useAutolineMatching": false,
      "multilineProcessingEnabled": false,
      "timeZone": "America/Los_Angeles",
      "automaticDateParsing": true,
      "forceTimeZone": false,
      "defaultDateFormat": "dd/MMM/yyyy HH:mm:ss"
    }
  ]
}
```

Remote File Source

Parameter	Type	Required?	Description
remoteHost	String	Yes	Host name of remote machine.
remotePort	Int	Yes	Port of remote machine (SSH)
remoteUser	String	Yes	User account to connect with the remote machine.
remotePassword	String	Yes	Password used to connect to remote machine. Required only when authMethod is set to "password".
keyPath	String	Yes	Path to SSH key used to connect to the remote machine. Required only when authMethod is set to "key".
keyPassword	String	No	Password to SSH key to connect to the remote machine, required only with authMethod is set to "password".

remotePath	String	Yes	Path of the file on the remote machine that will be collected.
authMethod	String	Yes	Authentication method used to connect to the remote machine. Options are "password" to connect with a password, or "key" to connect with an SSH key.
sourceType	String	Yes	Type "RemoteFile" .
name	String	Yes	Name of the Source. For example, "RemoteFile".
description	String	No	Type an optional description of the Source.
category	String	No	Describes the category type of the Source. For example, you could type "auth_logs" or "mail".
hostName	String	No	The host name of the Source.
timeZone	String	No	Type the time zone you'd like the Source to use in TZ database format. For example, America/Los_Angeles .
automaticDateParsing	Boolean	No	Determines if timestamp information is parsed or not. Type true to enable automatic parsing of dates (the default setting); type false to disable. If disabled, no timestamp information is parsed at all.
multilineProcessingEnabled	Boolean	No	Type true to enable; type false to disable. The default setting is true. Consider setting to false avoid unnecessary processing if you are collecting single-message-per-line files (for example, Linux system.log). If you're working with multi-line messages (for example, log4J or exception stack traces) then keep this setting enabled.
useAutolineMatching	Boolean	No	Type true to enable if you'd like message boundaries to be inferred automatically; type false to prevent message boundaries from being automatically inferred (equivalent to the Infer Boundaries option in the UI. The default setting is true.
manualPrefixRegex	String	No	Type a regular expression for the prefix of a message. Alternately, type "", to leave this empty.
forceTimeZone	Boolean	No	Type true to force the Source to use a specific time zone, otherwise type false to use the time zone found in the logs. The default setting is false .
defaultDateFormat	String	No	Type the default format for dates used in your logs. For more information about timestamp options, see Supported Time Stamp Conventions .
cutoffTimestamp	Long	No	This option allows you to specify the timestamp of the oldest logs you'd like to collect. Type the milliseconds since epoch of the oldest log.
cutoffRelativeTime	String	No	If you'd like to provide a relative cutoff of the logs you'd like to collect, type "-1h", to collect data that's been modified less than one hour old, "-1d", to collect data that's been modified less than one day old, or even "-1w", to collect data that's been modified less than one week old. Note that the

			parameter does not use the message timestamp, but instead uses the last modified timestamp.
filters	Boolean	No	If you'd like to add a filter to the Source, type the name of the filter ("Exclude", "Include", "Mask", or "Hash"). See the Collector Management API Read Me for more information.

Remote File Source JSON example:

```
{
  "api.version": "v1",
  "sources": [
    {
      "sourceType" : "RemoteFile",
      "name" : "SampleRemoteFile",
      "description" : "SampleRemoteFileSource",
      "category" : "events",
      "remoteHost": "myremotehost",
      "remotePort": 22,
      "remoteUser": "user",
      "remotePassword": "password",
      "keyPath": "",
      "keyPassword": "",
      "remotePath": "/var/log/somelog.log",
      "authMethod": "password",
      "timeZone": "America/Los_Angeles",
      "automaticDateParsing": true,
      "multilineProcessingEnabled": true,
      "useAutolineMatching": true,
      "manualPrefixRegex": "",
      "forceTimeZone": false,
      "defaultDateFormat": "dd/MMM/yyyy HH:mm:ss"
    }
  ]
}
```

Local Windows Event Log Source

Parameter	Type	Required?	Description
logNames	List	Yes	List of Windows log types to collect. For example, "Security", "Application", etc.
name	String	Yes	Name
sourceType	String	Yes	Type "LocalWindowsEventLog".
name	String	Yes	Type the name of the Source, like "SourceName",.
description	String	No	Type a description of the Source, like "SourceDescription",.
category	String	No	Type the description of the category of the Source. For example, you could type "auth_logs", r "mail",.
hostName	String	No	Type the host name of the Source, like "SourceHost",.
timeZone	String	No	Type the time zone you'd like the Source to TZ database format. For example, America/Los_Angeles .
sourceType	String	Yes	Type the correct type of Source (see specific Source types below for more information).
automaticDateParsing	Boolean	No	Determines if timestamp information is parsed or not. Type true to enable automatic parsing of dates (the default setting); type false to disable. If disabled, no timestamp information is parsed at all.
multilineProcessingEnabled	Boolean	No	Type true to enable; type false to disable. The default setting is true. Consider setting to false avoid unnecessary processing if you are collecting single-message-per-line files (for example, Linux system.log). If you're working with multi-line messages (for example, log4J or exception stack traces) then keep this setting enabled.
useAutolineMatching	Boolean	No	Type true to enable if you'd like message boundaries to be inferred automatically; type false to prevent message boundaries from being automatically inferred (equivalent to the Infer Boundaries option in the UI). The default setting is true.
manualPrefixRegexp	String	No	A type a regular expression for the prefix of a message. Alternately, type "", to leave this empty.
forceTimeZone	Boolean	No	Type true to force the Source to use a specific time zone, otherwise type false to use the time zone found in the logs. The default setting is false .
defaultDateFormat	String	No	Type the default format for dates used in your logs. For more information about timestamp options, see Supported Time Stamp Conventions .
cutoffTimestamp	Long	No	This option allows you to specify the timestamp of the oldest logs you'd like to collect. Type the milliseconds since epoch of the oldest log.
cutoffRelativeTime	String	No	If you'd like to provide a relative cutoff of the logs you'd like to collect, type "-1h", to collect data that's been modified less than one hour old, "-1d", to collect data that's been modified

			less than one day old, or even "-1w", to collect data that's been modified less than one week old. Note that the parameter does not use the message timestamp, but instead uses the last modified timestamp.
filters	Boolean	No	If you'd like to add a filter to the Source, type the name of the filter ("Exclude", "Include", "Mask", or "Hash"). See the Collector Management API Read Me for more information.

Local Windows Event Log JSON example:

```
{
  "api.version": "v1",
  "sources": [
    {
      "logNames": ["Security", "Application"],
      "sourceType" : "LocalWindowsEventLog",
      "name" : "LocalWinEventLogSource",
      "description" : "SampleLocalWinEventLogSource",
      "category" : "events",
      "timeZone": "America/Los_Angeles",
      "automaticDateParsing": true,
      "multilineProcessingEnabled": true,
      "useAutolineMatching": true,
      "manualPrefixRegex": "",
      "forceTimeZone": false,
      "defaultDateFormat": "dd/MMM/yyyy HH:mm:ss"
    }
  ]
}
```

Remote Windows Event Log Source

Parameter	Type	Required?	Description
domain	String	Yes	Windows domain from which logs will be created.
username	String	Yes	User name needed to connect to the remote machine.

password	String	Yes	Password needed to connect to the remote machine.
hosts	List	Yes	List of hosts to collect from.
sourceType	String	Yes	Type "RemoteWindowsEventLog" .
name	String	Yes	Name of the Source. For example, "RemoteWinEventsSource".
description	String	No	Type a description of the Source, like "SourceDescription",.
category	String	No	Type the description of the category of the Source. For example, you could type "auth_logs", r "mail",.
hostName	String	No	Type the host name of the Source, like "SourceHost",.
timeZone	String	No	Type the time zone you'd like the Source to use in TZ database format. For example, America/Los_Angeles .
automaticDateParsing	Boolean	No	Determines if timestamp information is parsed or not. Type true to enable automatic parsing of dates (the default setting); type false to disable. If disabled, no timestamp information is parsed at all.
multilineProcessingEnabled	Boolean	No	Type true to enable; type false to disable. The default setting is true. Consider setting to false avoid unnecessary processing if you are collecting single-message-per-line files (for example, Linux system.log). If you're working with multi-line messages (for example, log4J or exception stack traces) then keep this setting enabled.
useAutolineMatching	Boolean	No	Type true to enable if you'd like message boundaries to be inferred automatically; type false to prevent message boundaries from being automatically inferred (equivalent to the Infer Boundaries option in the UI). The default setting is true.
manualPrefixRegex	String	No	A type a regular expression for the prefix of a message. Alternately, type "", to leave this empty.
forceTimeZone	Boolean	No	Type true to force the Source to use a specific time zone, otherwise type false to use the time zone found in the logs. The default setting is false .
defaultDateFormat	String	No	Type the default format for dates used in your logs. For more information about timestamp options, see Supported Time Stamp Conventions .
cutoffTimestamp	Long	No	This option allows you to specify the timestamp of the oldest logs you'd like to collect. Type the milliseconds since epoch of the oldest log.
cutoffRelativeTime	String	No	If you'd like to provide a relative cutoff of the logs you'd like to collect, type "-1h", to collect data that's been modified less than one hour old, "-1d", to collect data that's been modified less than one day old, or even "-1w", to collect data that's been modified less than one week old. Note that the parameter does not use the message timestamp, but instead uses the last modified timestamp.

filters	Boolean	No	If you'd like to add a filter to the Source, type the name of the filter ("Exclude", "Include", "Mask", or "Hash"). See the Collector Management API Read Me for more information.
---------	---------	----	--

Remote Windows Event Log Source JSON example:

```
{
  "api.version": "v1",
  "sources": [
    {
      "domain": "mydomain",
      "username": "user",
      "password": "password",
      "hosts": ["myremotehost1", "myremotehost2"],
      "logNames": ["Security", "Application"],
      "sourceType" : "RemoteWindowsEventLog",
      "name" : "RemoteWinEventLogSource",
      "description" : "SampleRemoteWinEventLogSource",
      "category" : "events",
      "timeZone": "America/Los_Angeles",
      "automaticDateParsing": true,
      "multilineProcessingEnabled": true,
      "useAutolineMatching": true,
      "manualPrefixRegex": "",
      "forceTimeZone": false,
      "defaultDateFormat": "dd/MMM/yyyy HH:mm:ss"
    }
  ]
}
```

Syslog Source

Parameter	Type	Required?	Description
protocol	String	No	Protocol that syslog should use. Default is UDP; TCP is also supported.
port	Integer	Yes	Port that syslog should use to collect to the machine.
sourceType	String	Yes	Type "Syslog" .
name	String	Yes	Name of the Source. For example, "SyslogSource".
description	String	No	Description of the Source.

category	String	No	Describes the category type of the Source. For example, you could type "auth_logs" or "mail".
hostName	String	No	The host name of the Source.
timeZone	String	No	Type the time zone you'd like the Source to use in TZ database format. For example, America/Los_Angeles .
automaticDateParsing	Boolean	No	Type true to enable automatic parsing of dates; type false to disable. (The default setting is true.)
multilineProcessingEnabled	Boolean	No	Type true to enable; type false to disable. (The default setting is true.)
useAutolineMatching	Boolean	No	Type true to enable if you'd like message boundaries to be inferred automatically; type false to prevent message boundaries from being automatically inferred (equivalent to the Infer Boundaries option in the UI). The default setting is true.
manualPrefixRegex	String	No	A manually-entered regular expression for the prefix of a message.
forceTimeZone	Boolean	No	Type true to force the Source to use a specific time zone. (The default setting is false.)
defaultDateFormat	String	No	Type the default format for dates used in your logs. See Supported Time Stamp Conventions .
cutoffTimestamp	Long	No	This option allows you to specify the timestamp of the oldest logs you'd like to collect. Type the milliseconds since epoch of the oldest log.
cutoffRelativeTime	String	No	If you'd like to provide a relative cutoff of the logs you'd like to collect, type "-1h", to collect data that's been modified less than one hour old, "-1d", to collect data that's been modified less than one day old, or even "-1w", to collect data that's been modified less than one week old. Note that the parameter does not use the message timestamp, but instead uses the last modified timestamp.
filters	Boolean	No	If you'd like to add a filter to the Source, type the name of the filter ("Exclude", "Include", "Mask", or "Hash"). See the Collector Management API Read Me for more information.

Syslog Source JSON example:

```

{
  "api.version": "v1",
  "sources": [
    {
      "protocol": "UDP",
      "port": 514,
      "sourceType" : "Syslog",
      "name" : "SyslogSource",
      "description" : "SampleSyslogSource",
      "category" : "events",
      "timeZone": "America/Los_Angeles",
      "automaticDateParsing": true,
      "multilineProcessingEnabled": true,
      "useAutolineMatching": true,
      "manualPrefixRegexp": "",
      "forceTimeZone": false,
      "defaultDateFormat": "dd/MMM/yyyy HH:mm:ss"
    }
  ]
}

```

Script Source

Parameter	Type	Required?	Description
commands	List	Yes	List of command-line arguments.
file	String	No	Path to script file to run
workingDir	String	No	Working directory for commands/script.
timeout	Long	No	Script timeout (in milliseconds). By default, this is set to 0.
script	String	No	Script contents (if no file is provided).
cronExpression	String	Yes	Schedule for running the script. Must be a valid Quartz cron expression.
sourceType	String	Yes	Type "Script" .
name	String	Yes	Name of the Source. For example, "ScriptSource".
description	String	No	Type a description of the Source, like "SourceDescription",.
category	String	No	Type the description of the category of the Source. For example, you could type "auth_logs", r "mail",.
timeZone	String	No	Type the time zone you'd like the Source to use in TZ

			database format. For example, America/Los_Angeles .
automaticDateParsing	Boolean	No	Determines if timestamp information is parsed or not. Type true to enable automatic parsing of dates (the default setting); type false to disable. If disabled, no timestamp information is parsed at all.
multilineProcessingEnabled	Boolean	No	Type true to enable; type false to disable. The default setting is true. Consider setting to false avoid unnecessary processing if you are collecting single-message-per-line files (for example, Linux system.log). If you're working with multi-line messages (for example, log4J or exception stack traces) then keep this setting enabled.
useAutolineMatching	Boolean	No	Type true to enable if you'd like message boundaries to be inferred automatically; type false to prevent message boundaries from being automatically inferred (equivalent to the Infer Boundaries option in the UI). The default setting is true.
manualPrefixRegex	String	No	A type a regular expression for the prefix of a message. Alternately, type "", to leave this empty.
forceTimeZone	Boolean	No	Type true to force the Source to use a specific time zone, otherwise type false to use the time zone found in the logs. The default setting is false .
defaultDateFormat	String	No	Type the default format for dates used in your logs. For more information about timestamp options, see Supported Time Stamp Conventions .
cutoffTimestamp	Long	No	This option allows you to specify the timestamp of the oldest logs you'd like to collect. Type the milliseconds since epoch of the oldest log.
cutoffRelativeTime	String	No	If you'd like to provide a relative cutoff of the logs you'd like to collect, type "-1h", to collect data that's been modified less than one hour old, "-1d", to collect data that's been modified less than one day old, or even "-1w", to collect data that's been modified less than one week old. Note that the parameter does not use the message timestamp, but instead uses the last modified timestamp.
filters	Boolean	No	If you'd like to add a filter to the Source, type the name of the filter ("Exclude", "Include", "Mask", or "Hash"). See the Collector Management API Read Me for more information.

Script Source JSON Example:

```
{
  "api.version": "v1",
  "sources": [
    {
      "commands": ["/bin/bash"],
      "file": "/usr/local/bin/getlogs.log",
      "workingDir": "/var/log",
      "timeout": 60000,
      "script": "",
      "cronExpression": "0 * * * *",
      "sourceType" : "Script",
      "name" : "ScriptSource",
      "description" : "SampleScriptSource",
      "category" : "events",
      "timeZone": "America/Los_Angeles",
      "automaticDateParsing": true,
      "multilineProcessingEnabled": true,
      "useAutolineMatching": true,
      "manualPrefixRegex": "",
      "forceTimeZone": false,
      "defaultDateFormat": "dd/MMM/yyyy HH:mm:ss"
    }
  ]
}
```

Using Wildcards in Paths

Rather than entering each file by name, using wildcards in the Source path allows you to collect all files of a certain type within one or more directories, or many files from many directories. When specifying file names (or paths) in Microsoft Windows and Unix-like operating systems, the asterisk character (*) substitutes for any zero or more characters, and the question mark (?) substitutes for any one character.

Specifying Paths to collect from

When using wildcards in paths for file collections:

- * is a simple, non-recursive wildcard representing zero or more characters which you can use for paths and file names.
- ** is a recursive wildcard which can only be used with paths, not file names.

So, for example:

- /var/log/** will match all files in /var/log and all files in all child directories, recursively.
- /var/log/**/* .log will match all files whose names end in .log in /var/log and all files in all child directories, recursively.
- /home/*/.bashrc will match all .bashrc files in all user's home directories.
- /home/*/.ssh/**/* .key will match all files ending in .key in all user's .ssh directories in all user's home directories.



The recursive wildcard () can be specified only once in a path statement. It can be pretty confusing to use recursive logic at more than one level like in /var/log/**/subdir/**/* .log so we allow recursive wildcards in only one position in a path.**

Using Wildcards in the Blacklist Field

The same wildcards listed above can be used to exclude unwanted files or directories in the Blacklist field. For example, you are collecting /var/log/* .log but don't want to collect unwanted* .log, then specify /var/log/unwanted* .log in the Blacklist field. You can also exclude subdirectories. For example, if you are collecting /var/log/**/* .log but do not want to collect anything from /var/log/unwanted directory, specify /var/log/unwanted in the Blacklist field.



Sumo Logic does not collect any compressed files so you do not need to list them in the Blacklist field to exclude them.

CHAPTER 2

Using the Sumo Logic Web Application

The Sumo Logic Web Application connects you to your raw source data in the cloud. With a powerful and intuitive search capability, you can use the web application to expedite functions like forensic analysis, troubleshooting, and system health checks.

In this section, you will learn about:

- [The User Interface](#)
- [Checking the status of your Collectors](#)
- [Managing your deployed Collectors and their Sources](#)

Supported Browsers

The Sumo Logic Web Application can be run on any device with internet connectivity. Sumo Logic tests the Web Application on several browsers to ensure a consistent experience.

The following browsers are supported for running the Web Application:

- **Chrome** version 21 or higher
- **Firefox** version 14 or higher
- **Safari** version 5 or higher
- **Internet Explorer** version 9 or 10

Understanding the Web Application user interface

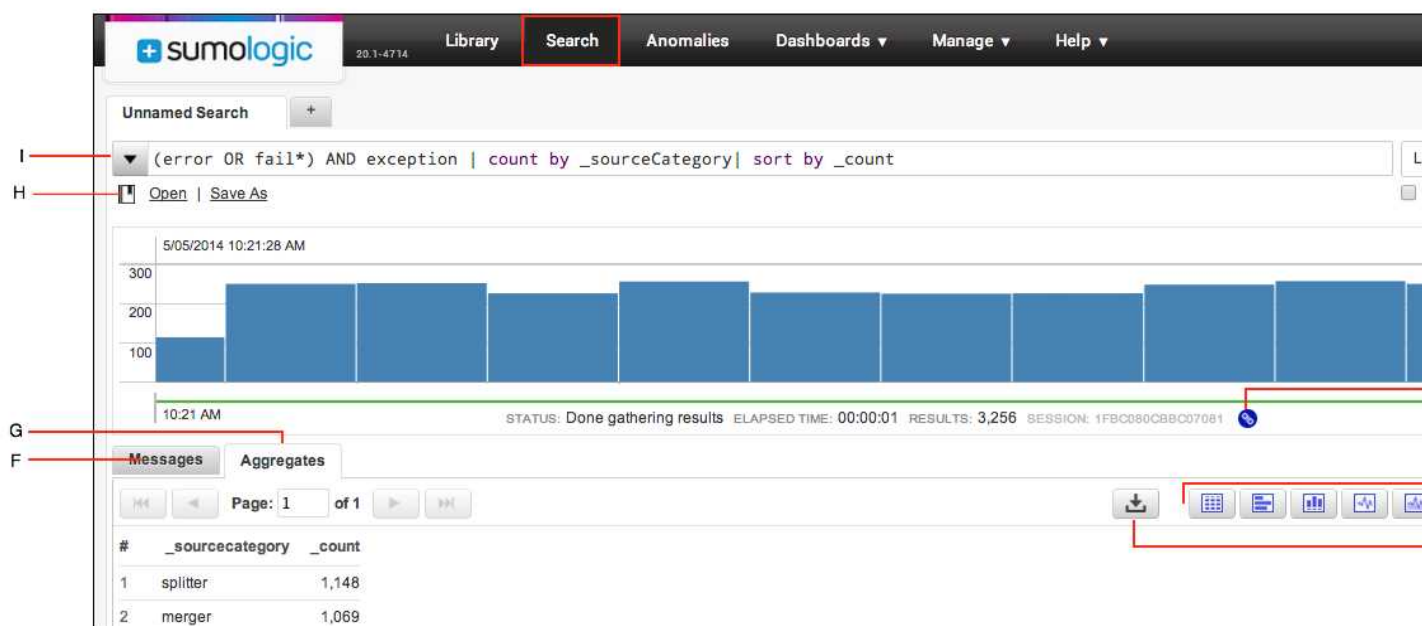
The **Sumo Logic Web Application** is made up of several pages that you can use to analyze your data, manage Collectors and Sources, view users in your organization, and otherwise keep an eye on your deployment.

Welcome

The **Welcome** page generally appears just the first time you log in to your account. Watch an overview video, use case videos, or browse through some common help and knowledge base topics.

Search

On the **Search** page, in the **Search** tab, you can enter simple or complex queries with time parameters to search your entire Sumo Logic data repository. You can select searches and run them from your Search Libraries. Your search results display in the **Messages** tab (for raw message data) or the **Aggregates** tab (for grouped results). You can run a saved search, pause, or stop searches, or schedule a search to run periodically and notify you of the results by email.



- A. Time range of the search.
- B. Send a link to the currently running search.
- C. Display options for search results.
- D. Collapse the top part of the Search page to view more results.
- E. Download and export search results (up to 10,000 records).
- F. View search results as messages.
- G. View aggregate search results.
- H. Open the Library.
- I. Search text box.

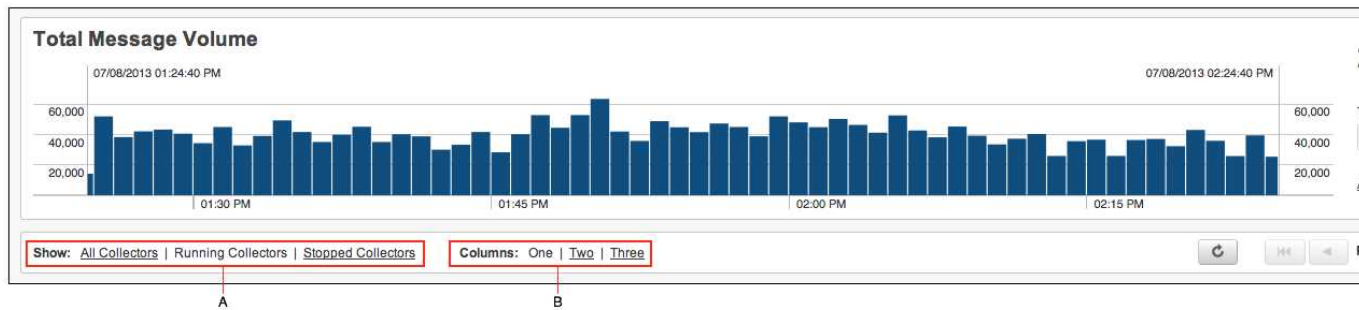


The maximum number of Search tabs that can be open at once is eight.

Status

The **Status** page provides a visual snapshot of the overall message history for your combined deployment of active Collectors. A histogram and message count for each Collector provides immediate feedback about traffic spikes or collection issues.

For more information, see [Checking Status](#).



- A. Choose an option to show or hide Collectors.
- B. Customize your view by choosing a Column option.
- C. Choose an option to limit the amount of time you'd like to see.

To view the **Status** page, select **Manage > Status**.

Collectors

From the **Collectors** page, you can manage all of your deployed Collectors and Sources. Select a Collector by name to add or edit its Source configurations. For information, see [Managing Collectors and Sources](#).

Manage Collectors and Sources

Upgrade Collectors Download Collector Ac

Show: All Collectors | Running Collectors | Stopped Collectors Expand: All | None

Page: 1 of 2

Name	Status	Source Category	Sources	Last Hour	Messages
▼ bill-1	✓		8		18,532
bill Local File	✓	bill			
boss Local File	✓	boss			
collector Local File	✓	collector			
collector_metrics_reporter Local File	✓	collector_metrics_reporter			
foo Remote File	✓				

Add Source

A

- A. Choose an option to show or hide Collectors.
- B. Choose an option to install, upgrade, or activate a Collector.
- C. Choose an option to add, edit, or delete a Source.

To view the **Collectors** page, select **Manage > Collectors**.

Users

From the **Users** page, Administrators can manage users and roles. For more information, see [Using RBAC with Sumo Logic](#).

Manage Users					View Roles	New User
Name	Email		Roles			
Amanda	amanda@demo.com	✓	Administrator	Edit		
Bill	bill@demo.com	✓	Analyst	Edit		
Brad	brad@demo.com	✓	Administrator	Edit		
Bobbie	bobbie@demo.com	⚠	Administrator	Edit		
Brigit	brigit@demo.com	✓	Administrator	Edit		
Bruno	bruno@demo.com	✓	Administrator	Edit		
Chase	chase@demo.com	✓	Administrator, Analyst	Edit		
Christian	daddy@demo.com	✓	Administrator	Edit		
Damon	damon@demo.com	✓	Administrator	Edit		
David	david@demo.com	✓	Administrator	Edit		

To view the **Users** page, select **Manage > Users**.

- A. Account details, including subscription information.
- B. Profile information.
- C. Account preferences.

Account

In the **Account** page provides an overview of your organization's account status.

sumologic

19.77-06

Search

Anomalies

Dashboards ▾

Manage ▾

Help ▾

Account

Basic

Data Management

Account Subscription

Your **Enterprise 1200** account has the following limits:
14.6 TB per month (500.0 GB / day average)
90 days of retention
1000 users

Billing

Usage over the current month

Month ends 04/01/2014 **3.9 TB** used of 14.6 TB
Average daily volume: **158.5 GB** / day

26.42%

Show Usage Details

Usage over the previous month

02/01/2014 - 03/01/2014 **4.0 TB** used of 14.6 TB
Total volume: **4.0 TB (142.6 GB)** / day average

27.56%

Show Usage Details

Preferences

The **Preferences** page is where you change your password and set other options for your personal account.

133

+

sumologic

20.1-1189

Search

Dashboards ▾

Manage ▾

Help ▾

Amanda (Demo) ▾

AccountSecurityPreferences

Preferences

My Profile

Organization: Demo

Username: amanda@demo.com

Password: [Change Password](#)

My Access Keys

Create

Label	Access ID	Type	Created	Status
loca1	suwweqxwrZUS77	Collector	11/01/2013	Active Delete

My Preferences

Web session timeout: 12 Hours ▾

* Any changes to the session timeout will take effect the next time you sign in.

☒ Automatically run the search after selecting it from a list of saved searches.

When editing queries:

☒ <Enter> runs the query, <Alt> <Enter> creates a new line.

☐ <Alt> <Enter> runs the query, <Enter> creates a new line.

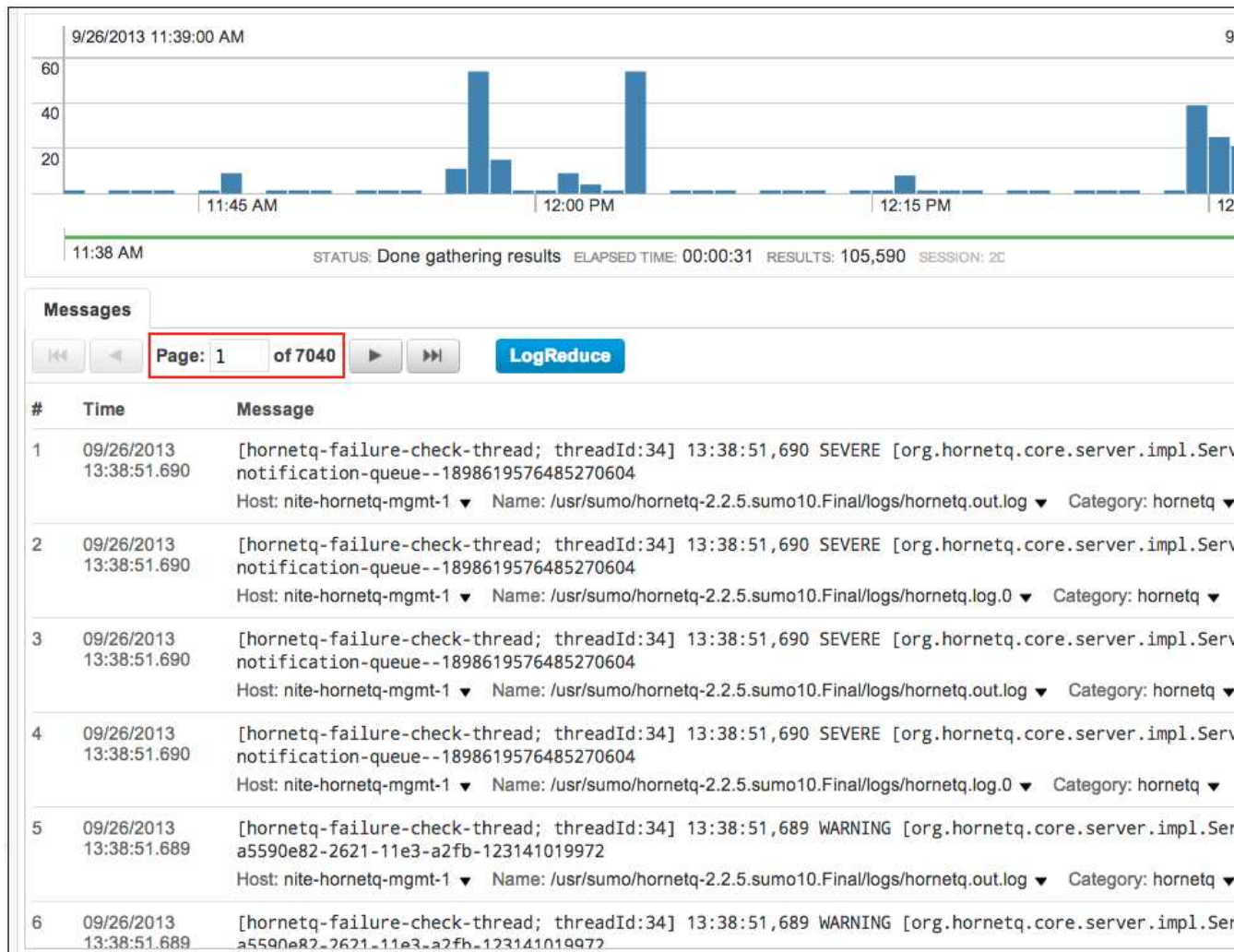
Save

Navigating through Messages in the Search Tab

When you run a search query, messages display in the Message, Aggregates, or Summarize tabs in the lower half of the browser window. Results are paginated with 15 messages per page.

If you have many pages of results, you have several options for navigation:

- Type a page number into the page number field and hit enter.
- For the Messages tab only, click into a block in the histogram to jump to the page containing the first message from that block. In the default sort order, the message is the most recent message from the block. In a reverse sort order, clicking into the histogram takes you to the page containing the oldest message from that block.



In this example, an unusual number of errors occurred between 12:00 and 12:15. Clicking into the tallest histogram block takes us to the page where logs related to the event can be viewed. This is the page that contains the most recent message from this message block. The pink vertical line moves along the histogram to show the approximate range in time that corresponds to the page you're viewing.

Preferences page

The **Preferences** page displays options you can use to configure the Sumo Logic Web Application.

The screenshot shows the Sumo Logic web application interface. At the top, there is a navigation bar with the Sumo Logic logo, a search bar, and links for Dashboards, Manage, and Help. On the right side of the navigation bar, the user's name 'Amanda (Demo)' is displayed with a dropdown menu. The dropdown menu has three options: 'Account', 'Security', and 'Preferences', with 'Preferences' being the selected and highlighted option.

The main content area is titled 'Preferences' and is divided into three sections:

- My Profile:** This section displays the user's organization as 'Demo', their username as 'amanda@demo.com', and a link to 'Change Password'.
- My Access Keys:** This section includes a 'Create' button and a table of existing access keys. The table has columns for Label, Access ID, Type, Created, and Status. One key is listed with the label 'loca1', Access ID 'suwweqxwrZUS77', Type 'Collector', Created '11/01/2013', and Status 'Active'.
- My Preferences:** This section allows the user to configure their session and search preferences. It includes a 'Web session timeout' dropdown set to '12 Hours', a note that changes take effect at the next sign-in, a checked checkbox for 'Automatically run the search after selecting it from a list of saved searches', and radio button options for 'When editing queries'.

Editing your Profile

Once a Sumo Logic account has been set up, the email address associated with the account cannot be edited. However, you can reset your password from the Preferences page. Instructions can be found in [Changing your password](#).

Creating and Managing Access Keys

Administrators are able to generate, activate, and delete Access Keys used to register Collectors with Sumo Logic. For more information see [Using Access Keys](#).

Editing your Preferences

Preference settings are only changed for your personal account; they don't affect any other users in your organization. Any changes you make to your preferences take effect the next time you sign in, not during the current session.

To change Preferences:

1. Set any of the following:

- **Web session timeout.** Choose an option to set the length of time before your Sumo Logic session times out.
 - **Automatically run the search after selecting it from a list of saved searches.** Keep this option selected if you'd like to run a saved search as soon as you select it. Deselect the option if you'd like to start the search manually.
 - **When editing queries.** Choose **<Enter> runs the query**, **<Alt> <Enter> creates a new line** or **<Alt> <Enter> runs the query, <Enter> creates a new line**, depending on the option you prefer.
2. Click **Save**.

Changing your password

You can change your password at any time through the Sumo Logic Web Application.

To change your password:

1. Click your name in the top menu bar.
2. Click **Preferences**.

The screenshot shows the Sumo Logic web application interface. The top navigation bar includes the Sumo Logic logo, a date/time indicator (19.07-43), and links for Library, Search, Anomalies, Dashboards, Manage, and Help. The user's name, Rosemary (Demo), is in the top right corner. A dropdown menu is open, showing 'Preferences' and 'Sign Out'. The 'Preferences' page is displayed, with the 'My Profile' section containing fields for Organization (Demo), Username (rosemary@demo.com), and Password (Change Password, highlighted with a red box). Below this is the 'My Access Keys' section with a 'Create' button. The 'My Preferences' section includes a 'Web session timeout' dropdown set to '15 Minutes', a note about session timeout changes, a checked checkbox for 'Automatically run the search after selecting it from a list of saved searches', and radio button options for query execution. A 'Save' button is located at the bottom right of the preferences section.

3. Click **Change Password**.
4. Enter your current password, and then enter the new password twice to verify it.

The screenshot shows a 'Change Password' dialog box. It has three input fields: 'Current Password', 'New Password', and 'New Password Again'. To the right of these fields is a list of password requirements, each preceded by a green checkmark: Minimum 8 characters, Maximum 32 characters, Lowercase letter (a-z), Uppercase letter (A-Z), Number (0-9), and Both passwords match. A 'Submit' button is located at the bottom right of the dialog box, highlighted with a red box.

5. Click **Submit** to finish resetting your password.

Web Application keyboard shortcuts

Shortcuts in the Web Application are triggered with a keystroke pattern of **g** and a second letter.

Shortcut	Action
g + s	Makes Search text box active. Hit Tab to use another shortcut.
g + a	Opens Anomaly Detection page.
g + l	Opens the Content Library.
g + c	Opens the Manage Collectors page.
g + t	Opens the Status page.
g+ u	Opens the Manage Users page.
g + x	Opens the Security page.
Shift + ?	Opens the list of shortcuts in the Web Application.

Search page keyboard shortcuts

The following shortcuts work only on the **Search** page.

Shortcut	Action
Alt + n	Opens a new search tab.
Alt + Shift + n	Opens a new search tab, prepopulated with the search being used in the current search tab.
Alt + q	Closes the current search tab.
Alt + <1-8>	Switches to the search tab, numbers one through eight.

CHAPTER 2

Searching and Analyzing

Sumo Logic's extensive query options help you gain valuable insight into your log messages. Sumo Logic collects and processes all your logs in real time so that you can search the most up-to-the-minute information:

- Use search engine-like syntax and quickly find records with relevant keywords.
- Save and share searches with others in your organization.
- Set up scheduled searches to receive results on a recurring basis.
- Configure Dashboards to keep an eye on search results in a graphical interface.
- Use Parser Libraries and other parse options to extract specific fields.



You'll find a number of webinars that focus on the different ways Sumo Logic can be used to search and analyze your data on the Sumo Logic website.

Running a basic search

After configuring Sources to collect the events and logs you need, you can begin using search within minutes. Sumo Logic search syntax uses logical and familiar operators allowing you to create ad hoc queries quickly and efficiently. You can save searches to re-use later or to run as regularly scheduled searches that can be delivered to your email address.

The basis of Sumo Logic Search Syntax is a funnel or "pipeline" concept. Beginning from all of your current Sumo Logic data, you enter keywords and operators separated by pipes ("|"). Each operator acts on the results from the previous operator so that you can progressively filter and pinpoint your search until you find exactly what you're looking for.

In the **Search** tab, a search query is typically formatted something like this:

keyword search | parse | where | group-by | sort | limit

Let's start with a basic search:

1. Sign into the **Sumo Logic Web Application**.
2. Click **Search** and enter a simple key term like "error" in the search field, or type an asterisk wildcard (*) to find all messages.



3. Hit **Enter** or click **Start**.
4. Sumo Logic returns all the log entries containing the search term in the Messages tab below the histogram.

Let's take a look at a slightly more complex search query to see how queries are formed. All queries begin with a keyword or string search. Wildcards are allowed including an asterisk (*) for zero or more characters and a question mark (?) for a single character. Strings can be parsed based on start and stop anchor points in messages and then aliased as user-created fields. All operators are separated by the pipe symbol (|).

Here's an example:

```
_sourcecategory=apache | parse "*" - "-" as src_ip | count by src_ip | sort _count
```

keyword expression
(in this case, a metadata field)

parse out the IP address into a field named "src_ip" using an endpoint anchor

count and sort the results

You can expand the complexity of your search queries with Sumo Logic search operators. Learn more about the basics of [Sumo Logic Search Syntax](#).

Keyword search expression

All search queries begin with a full-text search expression. For simplicity, we refer to this part of the search as a keyword search. Boolean logic and wildcards enable you to search for multiple terms, express logic about term distribution within messages, and specify partial terms with wildcards: use an asterisk (*), for zero or more characters, or a question mark (?) for a single character. With respect to Boolean syntax, the AND operator is implicit, meaning you do not need to type AND when entering multiple terms. Note that keyword searches are case-insensitive.

If you enter a phrase query such as an email or IP address, the Sumo Logic search engine looks for the individual indexed terms appearing next to each other. You can use a wildcard to represent a full term (This is allowed: `jsmith@*.com`), but not a partial term (This won't work: `jsmith@some*re.com`). Remember to enclose any keyword phrases containing spaces or special characters within quotes.

Everything you type in the query field before the first pipe is always a full-text search expression, even if it's just a wildcard (*) to find all results.

Some examples:

- *
- error OR exception OR fail* NOT debug
- `_sourceCategory="Cisco routers"`
- `("hr dept" "failed login") OR ("IT dept" "failed login")`

Sumo Logic operators

For detailed information about Sumo Logic Search syntax, see the [Search Syntax Reference](#). For a quick overview, here are the main Sumo Logic search operators with a syntax example for each:

- **Parse**
... | parse "start_anchor*stop_anchor" as fieldname | ...
Parse options include "parse anchor" as shown in the syntax example, or "parse regex" for using regular expressions to form more complex parse queries. It is acceptable to use "parse" for "parse anchor", or "extract" for "parse regex". For instructions on how to extract fields from messages using the parse operator, see [Parsing and Naming a Field](#).
- **Where**
... | where someField any_Boolean_operator someValue | ...
A conditional operator that can precede or follow another operator. Example combinations include "where x matches y", "where x in (a, b, c)", "where x not in (a, b, c)" and "where a > 1 and b / 4 < sqrt(x)".
- **If**
... | if(condition, value_if_true, value_if_false) as alias_field | ...
A ternary operator used to evaluate a condition as either true or false, with values assigned for each outcome. It is a shorthand way to express an if-else condition.
- **Summarize**
keyword expression | summarize
Summarize uses fuzzy logic and soft matching for pattern detection. The summarize operator automatically

groups messages by content similarity into clusters. For more information about the summarize algorithm, see [Detecting Patterns with Summarize](#).

- **Timeslice**

... | timeslice by time_period as fieldname

... | timeslice 25 buckets

Timeslice segregates search results by fixed time period, or by any number of buckets over a time range.

- **Group**

... | group-by-function (field_to_operate_on) group by (field_to_group_by) | ...

Group-by functions include count, count_distinct, sum, avg, stddev, max, min, last, and first. You can use "group" or "by" instead of "group by" so "count (*) group by user" is equivalent to "count by user". All group-by functions create a corresponding field preceded by an underscore, for example, _count.

- **Sort**

... | sort by sort_by_clause field_to_sort_by | ...

The Sort operator orders aggregate search results.

- **Limit**

... | limit 20

Use the Limit operator to reduce the number of messages or aggregated results returned.

- **Math Expressions**

... | expression as alias | ...

Supported mathematical operators include: +, -, *, /, %, as well as the ternary boolean operator "condition ? v0 : v1". A set of mathematical functions are also supported including abs, ceil, floor, round, exp, log, log10, pow, signum, sqrt, sin, cos, asin and acos.



If you have a set of messages displayed in the Messages tab, you can click any term within a message to add it to the query you are building (AND clicked_term). You can also use Alt-click to remove a term from results (!clicked_term, or NOT clicked_term).

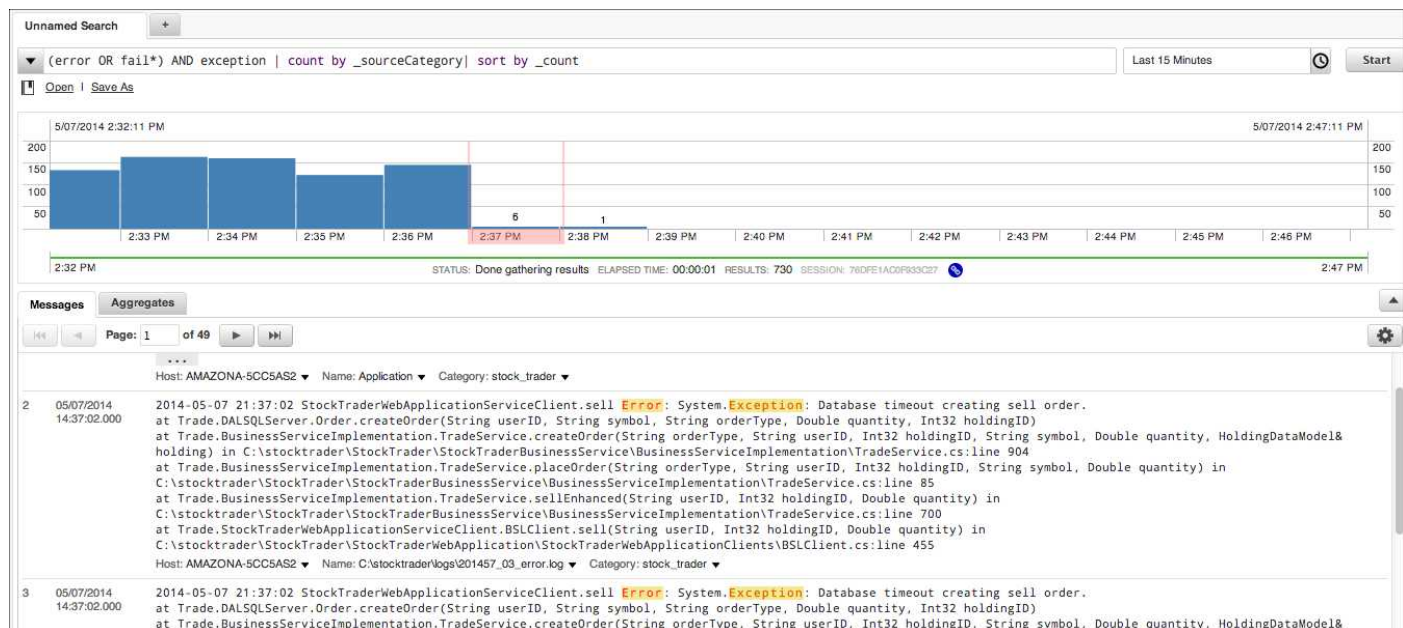
Search Highlighting

When you perform a search, and results are returned, your search terms are highlighted in the **Messages** tab.

For example, using this query:

(error OR fail*) AND exception | count by _sourceCategory | sort by _count

returns the following results in the **Messages** tab.



Note that the search terms **Error** and **Exception** are highlighted in yellow.

Highlighted search terms are limited to the first 1024 characters per message, in order to provide improved search performance.

Wildcards in Full Text Searches

For the most part, the asterisk (*) wildcard works the way you would expect in the keyword (full-text search) expression ("*" represents zero or more characters). This topic covers the details and special cases when using wildcards, for example, within quoted search expressions and within phrase queries.

Syntax:

- `fail*` matches fail, failure, fails, failed
- `e*or` matches error, extensor, eliminator, eor

Examples:

- `error OR fail error AND fail*`
- `(error OR fail) and debug error* OR (fail and debug) error? NOT fail`

Phrase Queries

A full-text search expression such as `jsmith@somewhere.com` is called a "phrase query" because it operates on multiple search terms: `jsmith`, `@`, `somewhere`, `.`, `com` as opposed to a single term query such as `"fail"`. In a phrase query, the Sumo Logic search language looks for the five terms in the email address adjacent to each other. So, you can search for things such as IP addresses or segments of IP addresses like `10.1.1.123`, or email domains like `@sumologic.com`.

If you are searching using a phrase query, Sumo Logic currently does not support partial-term wildcards within the phrase such as `jsmith@some*re.com`. However, you can use a wildcard to represent a full term in the phrase, like this `jsmith@*.com`.

Time Range Expressions

When you are building a search query, you have the option to add a **time range expression** in the separate **time range** field. Preset values are available to choose from, with **Last 15 Minutes** as the default start value and "now" being the implied end time. In cases where more control of the start and end time is required, you can type a time range expression directly into the time range field.



Relative Expressions

The Sumo Logic Web Application understands both absolute timestamps, as well as relative expressions.

Relative Expression	Definition
-1d	From one day (24 hours) ago to now.
-1d now	From one day ago to now.

-1d -12h	From one day ago to 12 hours ago.
-12h -60m	From 12 hours ago to 60 minutes ago.
-60m -600s	From 60 minutes ago to 600 seconds ago.

Either a single relative expression, or two relative expressions can be specified. If only one expression is present, it is interpreted as the start time, and the end time is automatically set to "now". The token "now" can be entered to mean the current time. If two expressions are present, the first one is interpreted as the start time, and the second one as the end time. Expressions should be prefixed with "-" to indicate that the time resolves to the past. The remainder of the expression contains a number, and a time multiplier. Valid multipliers are "s" for second, "m" for minute, "h" for hour, "d" for day.

Future time expressions (for example, **now to +15m**) are supported and will return results if timestamps for any collected data are set in the future.

Absolute Expressions

While relative expressions are useful, sometimes it is more important to express a specific point in time. If only one time expression is present, it is interpreted as the start time. If two expressions are present, the first one is interpreted as the start time, the second is interpreted as the end time. If only a date is entered, the time value is implied to be midnight. Again, the token "now" represents the current time. If no year is present in an absolute time expression, the current year is assumed.

US examples

Absolute Expression	Definition
04/01	From the most recent April 1st to now.
04/01/14 20:32:00 to 04/01/14 20:35:00	From April 1st, 2014 at 8:32 PM until April 1st, 2014 at 8:35 PM.
04/01 04/02	From midnight April 1st, 2014 to midnight April 2nd, 2014.
04/01/2014 00:00:00 to 04/02/2014	From midnight April 1st, 2014 to midnight April 2nd, 2014.
04/01/2014	From midnight April 1st, 2014 to now.
04/01/2014 04/02/2014	From midnight April 1st, 2014 to midnight April 2nd, 2014.

Worldwide examples

Absolute Expression	Definition
04.01	April 1st of the current year.
01-04-14 20:32:00 to 01-04-2014 20:35:00	From April 1st, 2012 at 8:32 PM until April 1st, 2012 at 8:35 PM.
01-04 02-04	From midnight April 1st, 2014 to midnight April 2nd, 2014.
01-04-2014 00:00:00 to 02-02-2104	From midnight April 1st, 2014 to midnight April 2nd, 2014.
01-04-2014	From midnight April 1st, 2014 to now.



There are many different ways to combine year, month, and days in an absolute time range expression. The Sumo Logic Web Application tries to interpret any input, up to the point where the time expression becomes ambiguous. For example, in the following expression, Sumo Logic cannot determine what is the year and what is the month:

11/02/01 11/04/02
04/01/11 04/02/11

To avoid indeterminate values, always enter the year in the format YYYY.

Saving a Search

Whether you are running ad hoc searches during a forensic investigation or running standard searches for health checks, you can save any search to run later. When you save a search, you have the option to set up the saved search to run at a scheduled interval with an automated notification by email of the search results. You can edit a saved search at any time.

To save a search:

1. In the Search tab, after typing your search query, click **Save As** below the search field.



2. For **Search name**, enter a name for your Saved Search. If you'd like, type an optional description to help you identify this search. (Optional.)
3. The search query populates automatically in the **Search** field. You can make changes to the search syntax or query details if you need to.
4. Choose a **Time Range** option that will be the default range when you run the saved search. If you'd like to search backwards, you can type a time range like -15m.
5. Choose a **Folder** to save your search. To add a new Folder to the Library, click the blue "+" and name the new folder.

Save Search

Search name*

Unnamed Search

Description

Search*

| parse "module=*j" as module
| timeslice 1m
| count as value by _timeslice, module

Timerange

Last 15 Minutes

Folder

PERSONAL

Run frequency

Never

At:

Timerange*

Alert condition

☒ Send notification every time upon search completion
☐ Send notification only if the condition below is satisfied:

Number of results

Equal to

Alert Type

Email

Recipients*

Separate email addresses with commas.

Cancel

Save

- If you'd like to set a schedule for this search, or to set the search to run periodically with an optional alert, set **Run Frequency** and **Alert Condition** options.
- Click **Save** to add the search to the **Library**.

Sharing a Search Link

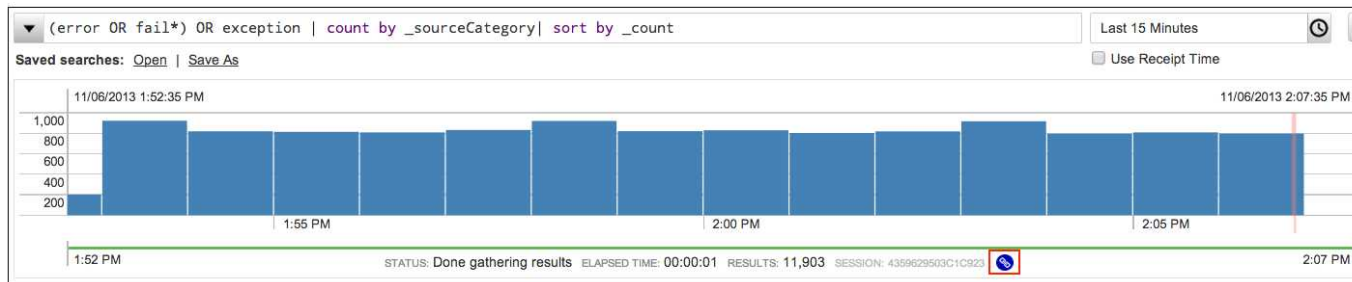
To share a search query with another user in your organization, you can send a Search Code. Click the **Share Search Code** icon to create a link that you can copy and paste. Then, the recipient can paste the link in the Search text box and run the search.



Search results may vary depending on each user's permissions.

To share a link to a search:

1. Run a search query you'd like to share.
2. After the search results are complete, click the **Share Search Code** icon under the results.



3. Do one of the following:
 - Copy and paste the top link to share via email or IM. Then the user can paste the link into a browser. (If the user is not currently logged in to Sumo Logic, he or she will be prompted to log in.)
 - If you know that the recipient is logged in to Sumo Logic, copy and paste the bottom code. This code can only be pasted into the Search text box in the Sumo Logic Web Application.

Share Search Code

Paste link in email or IM

https://sumologic.net/ui/#section/search/H5GEKD8JwRNsphYHd8dhKkfniDz5MAPMl

Paste code in the search query box

_code=H5GEKD8JwRNsphYHd8dhKkfniDz5MAPMuYXU3BE

Setting the Time Range of a Search

Enter time parameters for your search in the **time range** field, which is to the right of the Search field:

Unnamed Sear...

+

(error OR fail*) AND exception | count by _sourceCategory | sort by _count

Saved searches:

Open

|

Save As

02/26/2013 02:36:20 PM

8

6

4

2

02:40 PM

No search running currently.

Last 15 Minutes

This Hour

Today

Yesterday

This Month

Last Minute

Last 15 Minutes

Last 60 Minutes

Last 3 Hours

Last 6 Hours

Last 12 Hours

Last 24 Hours

Last 7 Days

Last 14 Days

Last 30 Days

All Time

You can choose from the preset time ranges (as shown above), or you can type a time expression using Sumo Logic time expression syntax. A few simple rules apply:

- For all time expressions with only one start time entered, the default assumed end-time is "now".
- If you enter two times, the first is assumed to be start time, and the second is assumed to be the end time for the range.
- When entering hours, you can use a 24 hour format, such as 17:32:00, or a 12 hour format, such as 5:32pm.
- To enter dates, always use a slash ("/") between month, day, and year in this format: mm/dd/yyyy. Sumo Logic does not support dates entered in yyyy/mm/dd notation or yy/mm/dd notation.
- If you enter a date, the assumed time for the date is midnight of that day. (00:00:00).
- For relative time, use these shorthand entries: d=day, h=hour, m=minute, and s=second.
- You can enter relative time ranges like this: "-1d -12h" for the range between one day ago and 12 hours ago.

Time range options may vary depending on the type of account your organization has. Sumo Logic Free accounts do not allow searches for longer than seven days.

Using Receipt Time

When collecting log data, the timestamp attached to messages is vital, both for the integrity of the data in your account, and for accurate query results. Because of the importance of timestamps, Sumo Logic indexes the timestamp of each message, making sure that data relevant to a query's time range is returned properly in search results, which allows you to reconstruct a correct event timeline.

150

To keep pace with real time analytics, Sumo Logic creates indices in nearly real-time. However, when collecting data with incorrect timestamps (or if there is latency in the collection of data), Sumo Logic can over-generate indices in attempt to properly handle the messages. These excess indices greatly degrade search performance, and this issue is referred to as index fragmentation. This can lead to the error message, "Your search contains messages that have been incorrectly parsed and cannot be displayed."

Sumo Logic has addressed index fragmentation by separating log messages into two categories:

- Messages with timestamps within plus or minus 24 hours from the present time.
- Messages with timestamps that fall outside this range (older than 24 hours from the present time, or in some cases, greater than 24 hours in the future).

During a typical search, only messages with timestamps within plus or minus 24 hours from the present time are queried. If a message's timestamp and receipt time don't match up, those messages may not be included in search results.

To search all data with any and all timestamps, select the **Use Receipt Time** check box. This option displays search results in reverse order of their receipt time, giving you the ability to view the difference in timestamp and receipt time to pinpoint Sources that may be generating incorrect timestamps.

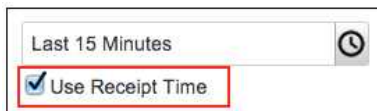
There are two instances when the Receipt Time option cannot be used:

- Scheduled searches cannot use Receipt Time.
- If you send a [link](#) to a search, the Receipt Time setting will not automatically be set for the person who uses the link to run a search.

Run a search by Receipt Time

To run a search by Receipt Time, select the **Use Receipt Time** check box:

1. Enter your query in the search text box.
2. Choose the Time Range for the query.
3. Select **Use Receipt Time**.

A screenshot of a search interface. At the top, there is a text input field containing 'Last 15 Minutes' and a clock icon to its right. Below this, there is a checkbox labeled 'Use Receipt Time' which is checked. The checkbox and its label are highlighted with a red rectangular border.

4. Review the search results for wide discrepancies between message timestamp and receipt time to pinpoint Sources with incorrect timestamps:

Messages			
<div> <div> <div>⏪</div> <div>⏴</div> </div> <div>Page: 1 of 676</div> <div> <div>⏵</div> <div>⏩</div> </div> <div>LogReduce</div> </div>			
#	Receipt Time	Time	Message
1	07/10/2013 10:42:26.920	07/10/2013 10:48:22.392	2013-07-10 17:48:22,392 Publishing message piles: '1', messages: '1'
2	07/10/2013 10:42:26.920	07/10/2013 10:48:22.000	[10/Jul/2013:17:48:22 +0000] libcurl/7.19.7 OpenSSL/0.9.8k zlib/1.2.3 HTTP/1.0" 200 192 "-" "curl/7.19.7 (x86_64-pc-linux-gnu)
3	07/10/2013 10:42:26.920	07/10/2013 10:48:21.391	2013-07-10 17:48:21,391 Publishing message piles: '1', messages: '1'
4	07/10/2013 10:42:26.932	07/10/2013 10:42:26.631	2013-07-10 10:42:26,631 [call=InboundSessionProtocol.cancelSession] [thread=Thread-568940 (group:HornetQ-client-global-threads-29210502)
5	07/10/2013 10:42:26.932	07/10/2013 10:42:26.567	2013-07-10 10:42:26,567 -0700 INFO [module=MIX] [localUserName=mix] [thread=Thread-568954 (group:HornetQ-client-global-threads-29210502)
6	07/10/2013 10:42:26.932	07/10/2013 10:42:26.535	2013-07-10 10:42:26,535 -0700 INFO [module=MIX] [localUserName=mix] [thread=Thread-568626 (group:HornetQ-client-global-threads-29210502)

Resolving timestamp/receipt time issues

If you notice an issue between timestamps and receipt time values, you can double-check the Source's settings. You can manually specify the parse format for the Source, and test the format to make sure it's valid. Learn more in [Setting Source timestamp options](#).

Alternately, if you're noticing that timestamps are parsing properly, check the timestamp conventions of your logs. Learn more in [Supported timestamp conventions](#).

Parsing and Naming a Field

The `parse` operator parses strings according to specified start and stop anchors, and then labels them as fields for use in subsequent actions such as sorting, grouping, or even math expressions.

Sample log message:

```
Jan 22 19:47:14 siminetids1 SFIMS: [1:200208:1] BLEEDING-EDGE Malware
Gator/Claria Agent Installed [Classification: Spyware] [Priority: 2]
{TCP} 178.16.44.117:4415 -> 10.152.73.153:80
```

To parse the type of classification out of this sample message (in this case, the word "Spyware"), write your query with start and stop anchors like this:

`snort | parse "[Classification: *]" as classID`

1. To choose a start anchor, find a point in the message characterized by a token or set of tokens that are unique to the message format. A token is any word, space, punctuation character, or adjacent set of alphanumeric characters. In the example above, the start anchor is a left bracket ([), the word **Classification**, a colon (:), and then a space character to match the exact syntax in the message.
2. Type an asterisk (*) after the start anchor without adding any additional spaces. The asterisk wildcard between the start and stop anchors is the glob representing the parsed term.
3. Type the stop anchor without adding any additional spaces or characters. The stop anchor is the next unique token after the parsed term. In the example above, the stop anchor is a right bracket (]).
4. Enclose the start anchor, the glob, and the stop anchor within one set of quotes.
5. After running a query with an aliased field, the field appears in the **Messages** or **Aggregates** tab in its own column.

#	Time	classid	Message
1	02/24/12 17:15:17.257	Detection of a Non-Standard Protocol or Event	Jan 17 09:00:28 SIMDMZIDS1 SFIMS: [116:58:1] snort_decoder: Experimental TCP options [Classification: Detection of a Non-Standard Protocol or Event] [Priority: 3] {TCP} 206.169.110.13 Host: sourcefire ▼ Name: syslog ▼ Category: ids ▼
2	02/24/12 17:15:12.250	Detection of a Non-Standard Protocol or Event	Jan 17 09:00:16 SIMDMZIDS1 SFIMS: [116:55:1] snort_decoder: Truncated Tcp Options [Classification: Detection of a Non-Standard Protocol or Event] [Priority: 3] {TCP} 10.209.7.8:8033 -> 10.136.193.70:62119 Host: sourcefire ▼ Name: syslog ▼ Category: ids ▼

6. Then to name the parsed field, type the word "as", and give the parsed field an alias that you can use downstream in the query for grouping, sorting, or other functions.



User-created (aliased) fields, such as extracted or parsed fields, can be named using alphanumeric characters and underscores (_). They must start with an alphanumeric character. Starting underscores are reserved for Sumo Logic fields, such as _sourceCategory.

7. In this example, the user-created fieldname is "classID". You could extend the query by adding any group-by function with the new field name, like this:

... | count classID | sort by _count



Remember that Sumo Logic automatically creates named fields for the output from group-by functions (such as _count, __timeslice, _count_distinct, _avg) and for metadata fields (such as _sourceCategory, _sourceHost, and others). Sumo Logic fields are always preceded by an underscore.

See Also

[Automatically Parsing Fields with Parser Libraries](#)

Pausing or Cancelling a Search

When a search is in progress, the options to **Cancel** or **Pause** the search appear.



Pausing a Search

Search always retrieves and displays messages in reverse chronological order. Results are found walking backward in time from your most current data and progressing through older data. So, if you pause a search, you can check the timestamps for the messages in the **Messages** tab. If you reverse the sort order of the messages so that the oldest message is at the top, then all messages with more recent timestamps have been retrieved and processed.

You can resume a paused search; just click **Resume** under the Start button.



Cancelling a Search

When you cancel a search, you are stopping all progress on the current search and removing all results. Your query remains in the search query field.

Modifying a Search from the Messages Tab

After running a search, you can modify subsequent searches by selecting text displayed in the **Messages** tab. After selecting text, you can choose how to modify the search using the options from a pop-up menu:

Messages

⏮

⏪

Page: 6

of 77

⏩

⏭

LogReduce

Host: nite-ftsearch-1

Name: /usr/sumo/search-20.1-514/logs/search.log

Category: search

84

03/04/2013 16:18:01.655

2013-03-04 16:18:01,655 -0800 INFO [hostId=nite-ftsearch-1] [module=STREAM] [1
[thread=flush-B81109DD3071D4BC] [auth=User:com@sumologic.com:000000000023A1ED:0
[call=InboundStreamProtocol.executeQuery] Calling search module with sessionId=
timeRange=1362442440000 - 1362442500000
Host: nite-ftsearch-1 Name: /usr/sumo/stream-20.1-515/logs/strea

Add the selected text as AND

Add the selected text as AND NOT

Add the selected text as OR

Add the selected text as OR NOT

Parse the selected text...

Depending on the option you select, the search is modified.

What's appended to my original search?

After you select text, one of the following options is added to your existing search.

Option	Added to Search
Add the selected text as AND	[search] AND [selected text]
Add the selected text as AND NOT	[search] AND ! [selected text]
Add the selected text as OR	[search] OR [selected text]
Add the selected text as OR NOT	[search] OR ! [selected text]
Parse the selected text...	[search] parse [selected text] as [fieldName]



After the option is added to your existing search, click **Start** (or press Enter/Return) to run the appended search.

Parsing a field from message text

If you come across text that you'd like to parse as a field, you can select that text and name the field from the **Messages** tab.

To parse a field from message text:

1. In the search results, select the text or string you'd like to parse, then click **Parse the selected text**.

Messages

Page: 1 of 127 **LogReduce**

#	Time	Message
1	03/04/2013 14:15:02.074	2013-03-04 14:15:02,074 -0800 INFO [hostId=nite-ftsearch-1] [module=SEARCH] [logger=scala.search.session.Search [thread=MTP-SearchQueryHandler-3] [auth=User:dickon@sumologic.com:000000000023A1ED:0000000000001891:false] [session=6EAE889F2CC30DE0] [customer=00000000000911891] [call=InboundSearchProtocol.startSearch] [session_path=A7 [explainPlan] exiting search query sessionId: '6EAE889F2CC30DE0'

- (Optional) In the **Parse Text** dialog box, select any text that you don't want to include in the parsed field. Then click **Extract this value**.
For example, to parse just the "customer" field, select the unique customer ID, then click **Extract this value**:

Parse Text

Select text that you want to parse.

customer=00000000000911891

Extract this value

Fields

Enter the field name(s), separated by comma.

- Type a name for the **Field**. This name appears at the top of the parsed column. (Field names can contain alphanumeric characters and underscores (_). The name must start and end with an alphabet character.) Then click **Submit**.

Note: Not entering a Field name will produce an error in the **Search** tab.

Parse Text

Select text that you want to parse.

customer=*

Fields

customerID

4. In the **Search** tab, click **Start** to begin the search.

Searching with LogReduce

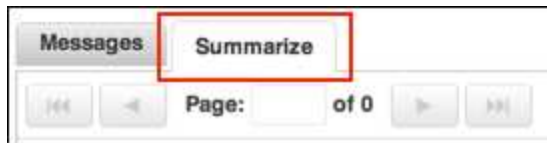
When you've already run a search query with non-aggregate results, you can use the **LogReduce** button in the **Messages** tab to automatically apply the **Summarize** operator to the current results.

To use the LogReduce button:

1. Run a search query with non-aggregate results.
2. In the **Messages** tab, the **LogReduce** button displays. Click it to automatically apply the Summarize operator to your results .



3. The **Summarize** tab is displayed with your results.



For information on how to use the **Summarize** operator and tab, see [Summarize Operator](#).

Metadata Searches

Sumo Logic attaches metadata tags to your log messages when the data is collected. Metadata tags are very useful when you're searching log data—they allow you to quickly identify the Collector or Source the data was ingested from, the type of logs, and so on. Some metadata fields are determined by the values you enter when installing a Collector or configuring a Source; these fields can be edited at any time.

What metadata fields can I search?

You can run queries using any of the following Sumo Logic metadata fields:

Name	Description
_collector	The name of the Collector (set when the Collector was installed) that received the log message.
_messageCount	A sequence number (per Source) added by the Collector when the message was received.
_messageTime	The timestamp of the message. If the message doesn't have a timestamp, messageTime uses the receiptTime.
_raw	The raw log message.
_receiptTime	The time the Collector received the message.
_size	The size of the log message.
_source	The name of the Source, determined by the name you entered when you <u>configured the Source</u> .
_sourceCategory	The category of the Source that collected the log message.

_sourceHost	The host name of the Source. For local Sources the name of the Source is set when you configure the Source . For remote Collectors, this field uses the remote host's name.
_sourceName	The name of the log file, determined by the path you entered when you configured the Source .

Searching metadata

To search using metadata:

1. As part of the keyword expression before the first pipe, enter the metadata type using its field name.
2. Add an equals sign (=).
3. Add the metadata string you want to search against. A few tips:
 - Add wildcards at the front and back of any partial term or string to capture the most results.
 - If your metadata contains spaces, quote the string and type the text exactly as entered at Source configuration time.
 - Quotes and wildcards cannot be used together.

Some examples and a description of each metadata type:

Example	Description
<ul style="list-style-type: none"> • _collector=Mac_server • _collector=AWS_1* 	Returns results from the named Collector only. Entered when a Collector is installed and activated.
<ul style="list-style-type: none"> • _source=main_web_app • _source=*syslog* 	Returns results from the named Source only. Entered when a Source is configured.
<ul style="list-style-type: none"> • _sourceCategory=*apache* • _sourceCategory="Security Logs" 	Returns results from one or more Sources depending on whether the tag was applied to a single Source or a series of Sources. Entered when a Source is configured.
<ul style="list-style-type: none"> • _sourceHost=hostname • _sourceHost=*RAS* 	<p>Usually returns results from one Source, unless a value is entered at the Collector level for a Collector with more than one Source.</p> <p>If the field is left blank when a Source is configured, the value for Source Host is taken from the host system value. A custom value can be entered at the Source or Collector configuration. Metadata values entered at Source level override Collector values.</p>
<ul style="list-style-type: none"> • _sourceName=path/to/file/ • _sourceName=*path* 	Returns results from one or more Source paths. Entered when a Source is configured. Note that the metadata field _sourceName is not the name of the Source, but the file path.

In the **Messages** tab, each message displays its metadata tags:

2012-04-16 11:33:33,152 -0700 INFO [hostId=nite-katta-2] [module=KATTA] [logger=net.sf.sessionID=2B83A38AA7E0D4AB, query=+(payload:error payload:fail*) +payload:exception +eve
Host: nite-katta-2 ▼ Name: /usr/sumo/katta-sumo/logs/katta-startNode.log ▼ Category: katta-slave ▼

Source Host
_sourceHost

Source Name
_sourceName

Source Category
_sourceCategory

For more information and suggestions on taxonomies, see [Establishing Metadata Conventions](#).

Searching Surrounding Messages

Surrounding messages allow you to investigate events surrounding a message from the context of the Host, file name or category identified enabling you to view the activity for the defined time period. As you browse results in the **Messages** list, you might come across a message where you would like to see more context: What other events occurred just before and after this event? What else was happening on this host at the same time? When you search surrounding messages, you capture the context of the current message to gain insight into surrounding activity.

After you launch a search on surrounding messages, the target message (the message from where your originated the search on surrounding messages) is highlighted in blue to help you keep your place.

To search surrounding messages:

1. For any message in the **Messages** tab, select the down-arrow next to one of the following:
 - **_sourceHost**. Matches messages based on the same system host.
 - **_sourceName**. Matches messages from the same file path AND the same host.
 - **_sourceCategory**. Matches messages based on the same user-created metadata.

Messages Aggregates

Page: 1 of 1

2012-06-14 09:15:18,607 -0700 ERROR [hostId=nite-frontend-1] [module=SERVICE] [logger=service.util.exception.Exception
Host: nite-frontend-1 ▼ Name: /usr/sumo/service-19.0-1626/logs/service.log ▼ Category: service ▼

Surrounding Messages > +/- 1 Minute
+/- 5 Minutes
+/- 10 Minutes

2. Select the time range to search before and after the selected message. Choose one minute, five minutes, or ten minutes. In this example, search will return messages for a ten minute time range (five minutes before, and five minutes after) from the same host and file path as the selected message.

Host: nite-frontend-1 ▼ Name: /usr/sumo/service-19.0-1626/logs/service.log ▼ Category: service ▼

Surrounding Messages > +/- 1 Minute
+/- 5 Minutes
+/- 10 Minutes

A new search tab opens displaying the surrounding messages. Your position in the log file is highlighted:

Messages			
Page: 97 of 200 Show Target Message			
Host: nite-frontend-1 Name: /usr/sumo/service-19.0-1626/logs/service.log Category: service			
1,445	06/14/2012 09:15:23.757	2012-06-14 09:15:23,757 -0700 INFO [hostId=nite-frontend-1] [module=SERVICE] [logger=service.end	
[remote_ip=71.204.167.46] [web_session=6x6kgek6...] Status for session: 3E1845A2B2C225EB, stream pendingWarnings=[], sessionIdString=3E1845A2B2C225EB}			
Host: nite-frontend-1 Name: /usr/sumo/service-19.0-1626/logs/service.log Category: service			
1,446	06/14/2012 09:15:23.757	2012-06-14 09:15:23,757 -0700 INFO [hostId=nite-frontend-1] [module=SERVICE] [logger=service.uti	
[remote_ip=71.204.167.46] [web_session=6x6kgek6...] Invocation: 'ReflectiveMethodInvocation: publ com.sumologic.service.endpoint.search.v2.api.SearchQueryService.getSearchQueryStatus(java.lang.St '*****',3E1845A2B2C225EB', elapsed: '0'			
Host: nite-frontend-1 Name: /usr/sumo/service-19.0-1626/logs/service.log Category: service			
1,447	06/14/2012 09:15:23.159	2012-06-14 09:15:23,159 -0700 ERROR [hostId=nite-frontend-1] [module=SERVICE] [logger=service.au	
'org.springframework.security.authentication.BadCredentialsException: Your login could not be ve			
Host: nite-frontend-1 Name: /usr/sumo/service-19.0-1626/logs/service.log Category: service			
1,448	06/14/2012 09:15:23.159	2012-06-14 09:15:23,159 -0700 WARN [hostId=nite-frontend-1] [module=SERVICE] [logger=service.end	
organization: 'FFFFFFFFFFFFFFFF' (msg: 'Login failed.', exc: 'Your login could not be verified')			
Host: nite-frontend-1 Name: /usr/sumo/service-19.0-1626/logs/service.log Category: service			
1,449	06/14/2012	2012-06-14 09:15:23,159 -0700 INFO [hostId=nite-frontend-1] [module=SERVICE] [logger=service.uti	

If you lose your place, you can always click **Show Target Message** to return to the highlighted message.



When you modify the results with surrounding messages, the search query is modified with the new time range and the host, path, or category appended to the keyword expression.

Exporting Search Results

After running a search query, you can download the results from your browser as a CSV (comma-separated values) text file.



The export from Sumo Logic is currently limited to **10,000 rows**.

If your organization has a **Sumo Logic Enterprise** account, and you'd like to export more than 10,000 rows, you can use the **Search Job API** to query Sumo Logic, then page through and output the results to a file of your choice. Learn more about the [Search Job API](#). (If the link does not open, right-click and open the link in a new browser window.)

To export grouped (aggregate) results:

1. Click the **Export Results** icon in the top-right corner of the **Aggregates** tab.



#	_sourcecategory	_count
1	receiver	952
2	boss	489
3	service	240
4	search	211
5	meta_metrics_reporter	146
6	config	142

2. Follow the prompts from your browser to download and save the file.

To export messages:

1. Click the **Preferences** icon in the top-right corner of the **Messages** tab, and then select **Export Results**.



#	Time	Message
1	04/04/2013 14:19:31.924	2013-04-04 14:19:31,924 -0700 ERROR [hostId=long-horne [logger=health.publish.LoggingHealthPublisher] [thread=receiver-output-search-queue_messageCount-7-thr receiver-output-search-queue_messageCount was over limi '300000' ms, last value was '222452' -> unhealthy: receiver-output-search-queue_messageCount was over limi '300000' ms, last value was '221302' Host: long-hornetq-inbound-1 Name: /usr/sumo/ops-19.36-5/logs/ops.log
2	04/04/2013 14:19:31.376	2013-04-04 14:19:31,376 [metrics-log-reporter] INFO com.sumologic.util.scala.MetricsReporter - Log 85505 (CALLS/MIN COUNT)

2. Click the **Download** link when it appears, and then follow any prompts from your browser to download and save the CSV file.



Messages

Page: 1 of 81

LogReduce

Export successful

Download

Graphing Search Results

In addition to the standard table view, you can view aggregated search results as a bar graph.

When graphing aggregate results from a query, the grouping function defines the plotted values on the one axis, and the grouping operator determines the values on the other axis. So, for example, **group by _sourceHost** produces a bar or point for each host. If you are using multiple group-by functions, a separate bar or point represents each set of grouped results.

To graph aggregate results:

- Click a graph button on the **Aggregates** tab:



Changing the scale to logarithmic

When graphing values that display over a wide range, change the scale to logarithmic to make the smaller values easier to perceive.

To change into or out of Logarithmic Scale:

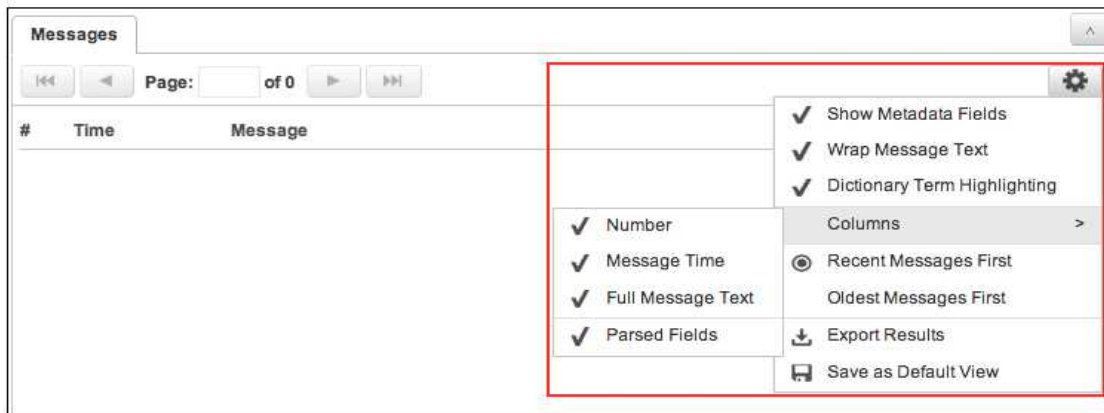
1. Click the button in the **Aggregates** tab to select the graph style you want.
2. Select the **Log Scale** check box to turn logarithmic scaling on. Deselect the check box to turn it off.



If your query does not produce graphable, numeric results, then the graphing buttons do not appear.

Setting Messages tab preferences

Preferences in the **Messages** tab allow you to customize how messages are displayed. You'll find the menu in the **Messages** tab. Just click the preferences icon:



To learn how to use the Export Results option, please see [Exporting Search Results](#).

To set Messages tab preferences:

1. Click the preferences icon in the **Messages** tab.
2. Set any of the following:
 - **Show Metadata Fields.** When selected, metadata field names are displayed below each log message:

#	Time	Message
1	04/03/2013 14:54:37.138	<p>[**] [1:254:4] DNS SPOOF query response with TTL of 1 min. and no authority [**] [Classification: Potentially Bad Traffic] [Priority: 2] 03/31-14:54:37.138341 172.16.0.23:53 -> 10.164.9.209:52804 UDP TTL:64 TOS:0x0 ID:3669 IpLen:20 DgmLen:89 Len: 61</p> <p>Host: nite-katta-1 ▼ Name: /var/log/snort/alert ▼ Category: security_logs ▼</p>

- **Wrap Message Text.** When selected, messages are wrapped to fit the browser window. If deselected you'll need to scroll to the right to read the full text of a message,
- **Dictionary Term Highlighting.** When selected, default Sumo Logic dictionary terms (such as error and exception) are displayed in red text:

1	02/07/2014 11:35:26.573	<p>2014-02-07 11:35:26,573 -0800 ERROR [hostId=long-receiver-2] [module=RECEIVER] [localUserName=receiver] [logger=scala.receiver.collector.CollectorStoreReceiverSourceManager] [thread=1684317026@qtp0-1551] Unable to get source for: 0000000000030A31 Reason: java.util.NoSuchElementException: Unable to get source for: 0000000000030A31 and its collector</p>
2	02/07/2014 11:35:26.324	<p>2014-02-07 11:35:26,324 -0800 ERROR [hostId=long-receiver-2] [module=RECEIVER] [localUserName=receiver] [logger=scala.receiver.collector.CollectorStoreReceiverSourceManager] [thread=1217932419@qtp0-1070] Unable to get source for: 0000000000030A31 Reason: java.util.NoSuchElementException: Unable to get source for: 0000000000030A31 and its collector</p>

- **Columns.** By default, the Number, Message Time, Full Message Text, and Parse Fields columns are displayed. If you'd like to remove one or more columns, deselect the column name.
 - **Recent Messages First/Oldest Messages First.** By default, **Recent Messages First** is selected, but if you'd prefer to view oldest log messages at the top of the Messages pane, select **Oldest Messages First**.
3. Click **Save as Default View** to retain the settings.

Writing Efficient Queries

An efficient search query returns targeted results as quickly as possible, with as little "noise" as possible. There are some easy ways to structure queries that run efficiently—returning better results in less time. To write more efficient search queries, see the following tips.



Searches automatically pause after 100,000 results are found. Using an efficient query can help you avoid this limitation.

Make the search as selective as possible

The more specific the query, the more efficiently it will run, as unnecessary messages are quickly thrown out of the mix. For example, the following two queries will generate the same result:

- `* | parse regex "uid=(?<userId>\d+)"`
- `"uid=" | parse regex "uid=(?<userId>\d+)"`

The second query will return the results more efficiently because the first query includes `"*"`, which prompts Sumo Logic to comb through all messages for the given time range.

Use the smallest parser library

Sumo Logic has several [parser libraries](#) that will speed up queries for specific log types by applying rules to evaluate messages according to pattern matches written as regular expressions.

Let's say you are parsing Apache Access logs. You can choose either:

- `* | parse using apache`

or

- `* | parse using apache/access`

The second query is more efficient because it doesn't waste time trying rules that don't apply, such as those that parse Apache Error logs.

Include the most selective filters first

It's best to filter data as early as possible in the query, using the most selective filters first.

For example, look at the following queries:

- `* | parse "queryTime=* " as queryTime | parse "uid=* " as uid | where queryTime > 10000`
- `* | parse "queryTime=* " as queryTime | where queryTime > 10000 | parse "uid=* " as uid`

Because most log lines have a `uid`, but only a small fraction have `queryTime > 10000`, the second query is more efficient.

Parsing

Sumo Logic provides a number of ways to parse fields in your log messages:

Parse operator. Automatically parses and names a field.

Parse regex. Parse regex (or "extract") enables users comfortable with regular expression syntax to extract more complex data from log lines. Parse regex can be used, for example, to extract nested fields.

Parse nodrop. For all parse operators, messages must match at least one segment of the parse expression or they are dropped from the results. Adding the "nodrop" option forces results to also include messages that do not match any segment of the parse term.

Parser Libraries. Each Parser Library is a set of parsing definitions written to automatically extract field values from a specific type of raw data. Rather than writing field parsing rules yourself, you can apply a parser library to your search to automatically extract fields from the logs, and then use the resulting parsed fields to extend your query.

Parse Operator

The **parse** operator (also called the parse anchor) parses strings according to specified start and stop anchors, and then labels them as fields for use in subsequent aggregation functions in the query such as sorting, grouping, or other functions. For help with learning to parse strings, see [Extracting and Naming a Field](#).



User-created fields, such as extracted or parsed fields, can be named using alphanumeric characters as well as underscores ("_"). Fields must start with an alphanumeric character.

Syntax:

- ... | parse "start_anchor*stop_anchor" as fieldname | ...
- ... | parse "start_anchor*stop_anchor" as fieldname no drop| ...

Rules:

- Fieldname can use underscores, but must start with an alphanumeric character.
- Supports wildcard * for anchor text. Two wildcard characters cannot appear next to each other, so " **" is not valid.
- The number of wildcards in the pattern string must match the number of variables.
- Multiple expressions are allowed for a single parse operator.
- If you add the option **nodrop**, then messages that match zero terms in the expression are included in the output. (See the section on [Parse ... nodrop](#) for a more thorough explanation of the **nodrop** option)
- Can be used with **parse regex** operator.

Examples:

Sample log message:

```
Aug 2 04:06:08: host=10.1.1.124: local/ssl2 notice mcpd[3772]:  
User=jsmith@demo.com: severity=warning: 01070638:5: Pool member  
172.31.51.22:0 monitor status down.
```

In the following examples, the start_anchor is "user=" and the stop_anchor is ":", which ends the email address. The asterisk (*) is the glob representing the parsed term. The examples create a new field for each message named "user" and that field will contain the value of the email address, in this case `jsmith@demo.com`.

- ... | parse "user=*" as user | ...
- ... | parse "user=*" as user nodrop | ...
Includes messages that do not match the initial query.
- status AND down | parse "user=*" as user | ...

The parse operator also allows you to extract multiple fields in one command:

- status AND down | parse "user=: severity=*" as (user, severity) | ...
This example creates two fields from the sample log message: `user=jsmith@demo.com` and `severity=warning`.

Parse Regex or Extract Operator

The **Parse Regex** operator (also called the extract operator) enables users comfortable with regular expression syntax to extract more complex data from log lines. Parse regex can be used, for example, to extract nested fields.



User added fields, such as extracted or parsed fields, can be named using alphanumeric characters as well as underscores ("_") and dashes ("-"). They must start and end with an alphanumeric character.

Syntax

- ... | parse regex "start_anchor_regex(<field_name>.*?)stop_anchor_regex" | ...
- ... | parse regex "start_anchor_regex(<field_name>.*?)stop_anchor_regex" nodrop | ...
- You can use the alternate term "extract".
- For more information, see [regular expressions](#).

Rules

- Regex must be a valid Java regular expression enclosed within quotes.
- Matching is case sensitive. If any of the text segments cannot be matched, then none of the variables will be assigned.
- Multiple parse expressions are processed in the order they are specified. Each expression always starts matching from the beginning of the message string.
- Multiple parse expressions can be written with shorthand using comma-separated terms.
- Add the option **nodrop** to allow messages that match zero terms in the expression to be included in the output.
- Can be used with the parse operator.

Examples

Parsing an IP address

Extracting IP addresses from logs is straight-forward using a parse regex similar to:

```
... | parse regex "(?<ip_address>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})" | ...
```

Parsing multiple fields in a single query

Parse regex supports parsing out multiple fields in one query. For example, say we want to parse username and host information from logs. Use a query similar to:

```
... | parse regex "user=(?<user>.*?):" | parse regex "host=(?<msg_host>.*?):" | ...
```

Sub-parsing a previously parsed field

In our example, we parsed a field named `user_agent` that contains information about browsers attempting to access a website. Now we want to create a new field named `mobile_device` (derived from `user_agent`) that contains an iPhone string. Use a query similar to:

```
_sourceCategory=*apache* | parse using public/apache/access | parse regex field=user_agent ".+?(?<mobile_device>iPhone);"
```

Indicating an OR condition

In situations where we want to use an OR condition, where we have multiple possibilities that may match the regular expression, the best practice is to use the **? operator**. For example, say we're looking for either `string1`, `string2`, or `string3`. Use a query similar to:

```
... | parse regex .... (?<string1|string2|string3| ...)
```

For a more concrete example, if we have a field named `URL` and we want to create a new field that includes domains that end with `.com`, `.co.uk`, or `.com.au`, search with a query similar to:

```
... | parse regex field=url "[0-9A-Za-z-]+\.(?<domain>[A-Za-z-]+\.(?:co.uk|com|com.au)).**"
```

For a more complex example, see [Refining a Query](#).

Extracting multiple values for a single field

In addition to parsing a field value, the **multi** option allows you to parse multiple values *within* a single log message. This means that the **multi** keyword instructs the parse regex operator to not just look for the first value in a log message, but for all of the values, even in messages with a varying number of values. As a part of this process, the **multi** keyword creates copies of each message so that each individual value in a field can be counted.

For example, say our firewall log messages look like this:

```
http://www.bigdata.biz" "CU1_4919|967:925:123
```

From this message, we'd like to extract the firewall codes. Use the **multi** keyword in the parse regex:

```
... parse url first  
| parse regex "Firewall Rules: \|(?<trigger_rules>.*?)\|"  
| parse regex field=trigger_rules "(?<trigger_rule>\d+)" multi
```

The output looks like:

Messages				
Page: 1 of 1 LogReduce				
#	Time	trigger_rules	trigger_rule	Message
1	04/16/2013 11:15:05.948	967:925:123	967	M1 Firewall Rules: 967:925:123 Host: rishis-macbook ▼ Name: /private/var/log/test/test.log ▼ Category: rd_test ▼
2	04/16/2013 11:15:05.948	967:925:123	925	M1 Firewall Rules: 967:925:123 Host: rishis-macbook ▼ Name: /private/var/log/test/test.log ▼ Category: rd_test ▼
3	04/16/2013 11:15:05.948	967:925:123	123	M1 Firewall Rules: 967:925:123 Host: rishis-macbook ▼ Name: /private/var/log/test/test.log ▼ Category: rd_test ▼

As each value has its own message, you can use any of the parsed values in an aggregation.

Parse nodrop Option

For all parse operators, messages must match at least one segment of the parse expression or they are dropped from the results. Adding the **nodrop** option forces results to also include messages that do not match any segment of the parse term.

Syntax:

- * | parse "a=*" as a nodrop
- * | parse "a=*" as a nodrop | parse "b=*" as b
In this case, messages that match either a or b are output.
- * | parse "a = *" as a | parse "b = *" as b
In this case, both parse operators are implicitly dropping non-matching messages. This means only messages that match both a and b are output.
- * | parse "a=*" as a nodrop | parse "b=*" as b nodrop | parse "c=*" as c nodrop | parse "d=*" as d
In this case, messages that match (a or b or c or d) are output. Everything else is dropped.

Rules:

- Messages with zero matches are included in the output but do not contain any ALIAS fields or tags related to the parse expression.
- Using the nodrop option, you can express advanced Boolean logic in choosing your desired message output when you chain the Parse operators.

Examples:

Use the nodrop option with a parser

Queries can use the nodrop option with a parser, such as the Apache Access parser:

```
_sourceCategory=Apache* | parse using public/apache/access nodrop
```

Use the nodrop option with parse regex

You can parse out an IP address using parse regex and parse nodrop:

```
_sourceCategory=Apache* | parse regex "(?<src_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})" nodrop
```

Use parse nodrop as an OR condition

For example, you can either GET or POST prefix before the URL in this query:

```
_sourceCategory=Apache* | parse "GET * HTTP" as url nodrop | parse "POST * HTTP" as url
```

Extracting Fields with Parser Libraries

To expedite searching and analysis for common log types, Sumo Logic offers **Parser Libraries**. Each Parser Library is a set of parsing definitions written to automatically extract field values from a specific type of raw data. Rather than writing field parsing rules yourself, you can apply a parser library to your search to automatically extract fields from the logs, and then use the resulting parsed fields to extend your query.

Currently Sumo Logic offers Parser Libraries for the following:

- [Apache Access Logs](#)
- [Apache Error Logs](#)
- [Cisco ASA Logs](#)
- [Microsoft IIS Logs](#)

Using Parser Libraries

So, for example, if you are collecting Apache logs and you want to automatically parse common fields from those logs, you can apply the Apache Parsing Library to your search.

In the **Messages** tab, the parsed field values appear in labeled columns:

Messages								
Page: 1 of 13								
#	Time	referrer	size	src_ip	status_code	url	user_agent	Message
1	03/05/12 13:27:41.0000	http://www.freessoft.org /CIE/Course/Section3 /7.htm	24308	217.17.33.206	200	/CIE/Course /Section3/7.htm	Mozilla/4.0 (compatible; MSIE 5.5; Windows 95)	217.17.33.206 - - /CIE/Course/Section3/7.htm "http://www.freessoft.org/ Mozilla/4.0 (com Host: www.freessoft.org Name: /var/www/vhost Category: apache ▼
2	03/05/12 13:27:32.0000	http://www.freessoft.org /CIE/RFC/2131/22.htm	10174	100.43.83.161	200	/CIE/RFC /2131/23.htm	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)	100.43.83.161 - - /CIE/RFC/1583/45. "Mozilla/5.0 (com

Leveraging Parsed Fields in Searches

To invoke the parser, use the **parse using** operator in your search query like this:

```
keyword expression | parse using public/name_of_parsing_library
```

When you apply a parser library to your search, your messages are parsed into named fields automatically. You can extend your query using the named fields. For example, if you were investigating the top URLs served from your website by byte size, you could use this query:

```
_sourceName=*access+log* AND _sourceCategory=*apache* | parse using public/apache/access | avg(size) as average_size_KB by url | sort by average_size_KB | limit 100 | (average_size_KB/1000) as average_size_KB | (toLong(average_size_KB*100)/100) as average_size_KB
```

In the example above, the **size** field is automatically parsed and named so the query can begin by calculating the average size based on the **size** field. Without using the parser, you would first need to manually extract and name the **size** field using a **parse anchor** or **parse regex** expression.

Understanding Parser Hierarchies

The Parser Libraries are arranged as hierarchical sets that include more specific subsets. When you invoke the parser, you can choose the level of generality or specificity to apply to your search. For example, the parent Apache parser currently contains two child subsets: one for parsing Apache error logs and one for parsing Apache access logs.



Apache 2.2 refers to parsing rules relevant to logs coming from Apache v2.2. Apache 2.4 refers to parsing rules relevant to logs coming from Apache v2.4.

Let's look at some examples. Using the Apache Parser Library, you can invoke the entire set of parsing definitions like this:

```
<keyword expression filtering for Apache logs> | parse using public/apache
```

Since the global Apache parser includes definitions for 14 named fields, your results will show 14 labeled columns in the **Messages** tab.

It's recommended that you write a query to invoke a more specific subset of definitions; the more specific parser library definition you invoke, the faster you'll get results. For example, to parse log messages relevant to Apache version 2.4, use the following query:

```
<keyword expression filtering for Apache errors> | parse using public/apache/error/2.4
```

Since the more specific Apache errors version 2.4 parser includes definitions for only three fields, your results will show three labeled columns in the Messages tab.



The Sumo Logic Parser Library is divided into two sets: public and organization-specific. The public parsers are available to everyone. The organization-specific parsers are private and only available to users within an organization. If you would like Sumo Logic to build an organization-specific parser library for your logs, contact your Technical Sales Representative or [file a Support request](#).

Understanding Matching

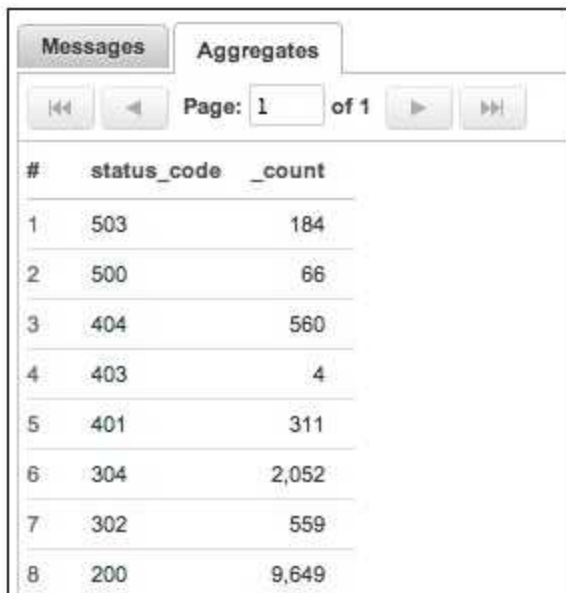
To apply the **apache** set of parsing rules to your search:

1. In the search query field, type any keyword expression to filter on your Apache logs (or type an asterisk [*] wildcard).
2. Then type a pipe symbol (|) and the **parse using** operator followed by the path to the parser (public/apache).
3. Extend your query using operators such as **where** or **sort** with the automatically parsed fields (src_ip, url, referrer, status_code, size, or user_agent).

For example, this search:

```
_sourceCategory=apache | parse using public/apache/access | count by status_code | sort by status_code
```

yields these results (plus additional fields) with no manual parsing required since the field **status_code** is parsed for you:



The screenshot shows a search interface with two tabs: 'Messages' and 'Aggregates'. The 'Aggregates' tab is active, displaying a table with three columns: '#', 'status_code', and '_count'. The table contains 8 rows of data. Above the table, there are navigation controls including 'Page: 1 of 1' and arrows for navigating between pages.

#	status_code	_count
1	503	184
2	500	66
3	404	560
4	403	4
5	401	311
6	304	2,052
7	302	559
8	200	9,649

Notice that the first group of messages (#1) in the example above has an empty **status_code** column. When using parser libraries, messages that match any single parsing definition (not all parsing definitions) are included in results. So, in the example above, 12 messages matched at least one parsing definition, but did not include status codes.

Parsing Apache Logs

This parser library processes the logs from Apache web servers in the NCSA extended/combined log format.

This table shows the global set of parsed Apache fields and the subset paths of the **public/apache** parser library:

Field Name	Description	Apache Error	Apache Access	Apache Error 2.2	Apache Error 2.4
file	The file name.	✓		✓	✓
log_level	The log level such as error, warn, crit, emerg, alert.	✓		✓	✓
method	The HTTP request method.		✓		
module	The Apache module involved.	✓			✓
process_id	The operating system process id.	✓			✓
reason	The reason for the error.	✓		✓	✓
referrer	Referrer URL.	✓	✓	✓	✓
request	The client request.	✓		✓	✓
size	Size of the response.		✓		
src_ip	Source IP address of the requesting client.	✓	✓	✓	✓
src_port	The client request port.	✓			✓
status_code	The HTTP status code.		✓		
thread_id	The operating system thread ID.	✓			✓
url	URL of the request.	✓	✓	✓	✓
user_agent	The user agent string.		✓		

To limit the fields parsed to a subset (access, error, or error version), apply the more specific parser path in your query. For example, to parse and output the seven Apache access log fields, use the path to the access log subset like this:

```
keyword expression | parse using public/apache/access
```

To parse log messages relevant to Apache version 2.4, use the following query:

```
keyword expression | parse using public/apache/error/2.4
```

To capture all messages, even if they match zero parsing definitions, add the **nodrop** option to your query, like this:

```
_sourceCategory=apache | parse using public/apache nodrop | count by status_code | sort by status_code
```


Parsing Apache Access Logs

The Apache access log parser extracts and labels the following fields.

Field Name	Description
method	The HTTP request method (GET, POST, etc.).
referrer method	Referrer URL.
size	Size of the response.
src_ip	Source IP address of the client requesting the resource.
status_code	HTTP status code.
url	URL of the request.
user_agent	The user agent string.

To apply the Apache Access Log parser to your search:

1. In the search query field, type any keyword expression to filter on your Apache logs (or type an asterisk [*] wildcard).
2. Then type a pipe symbol (|) and parse using **public/apache/access**.
3. Extend your query using operators such as **where** or **sort** with the named fields (src_ip, url, referrer, status_code, size, or user_agent).

In the **Messages** tab, the parsed field values appear in labeled columns:

Messages									
Page: 1 of 3									
#	Time	method	referrer	size	src_ip	status_code	url	user_agent	
1	05/07/2012 10:06:32.000	GET	http://twitter.com	30372	199.83.131.129	200	/news-and-events/	Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.3) Gecko/20100405 Namoroka/3.6.3	
2	05/07/2012 10:06:32.000	GET	http://www.sumologic.com/company/	5153	199.83.131.129	200	/company/careers/	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:11.0) Gecko/20100101 Firefox/11.0	
3	05/07/2012 10:06:32.000	GET	http://www.sumologic.com/product/security/	4434	199.83.131.129	200	/product/pricing/	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/19.0.1084.30 Safari/536.5	

Parsing Apache Error Logs

The Apache error log parser extracts and labels the following fields.

Field Name	Description	Apache Error 2.2	Apache Error 2.4
file	The file name.	✓	✓
log_level	The log level such as error, warn, crit, emerg, alert.	✓	✓
module	The Apache module involved.		✓
process_id	The operating system process ID.		✓

Field Name	Description	Apache Error 2.2	Apache Error 2.4
reason	The reason for the error.	✓	✓
referrer	Referrer URL.	✓	✓
request	The client request.	✓	✓
src_ip	Source IP address of the requesting client.	✓	✓
src_port	The client request port.		✓
thread_id	The operating system thread ID.		✓
url	URL of the request.	✓	✓

To apply the Apache error log parser to your search:

1. In the search query field, type any keyword expression to filter on your Apache logs (or type an asterisk [*] wildcard).
2. Then type a pipe symbol (|) and **parse using public/apache/**.
 - To target the output to Apache version 2.2 fields, type **parse using public/apache/error/2.2**
 - To target the output to Apache version 2.4 fields, type **parse using public/apache/error/2.4**
3. Extend your query using operators such as **where** or **sort** with the named fields, like this:
(error OR fail*) AND exception | parse using public/apache/error | sort by log_level

If you know the version of Apache you are using, we recommend using the parser built for your specific version.

Parsing Cisco ASA Logs

The Cisco ASA parser assumes that interfaces, IP addresses, and port tuples are either "Source" or "Destination" tuples, and categorizes each according to the flow specified in the message. The Source interface is assumed to be the default; the Source Host is assumed to be the default host.

The Cisco ASA parser extracts and labels the following fields:

Field Name	Description
access_decision	Specific to a message (%ASA-4-106100) that appears frequently; denotes whether a packet was permitted or denied.
acl_id	ID of the access control list that caused the log message.
action	A generic field that denotes the action or what the message actually means; wherever possible the specific value is extracted.
dest_host	The destination host (either an IP address or string) denoting the hostname.

Field Name	Description
dest_interface	The destination interface.
dest_port	The destination port.
hit_cnt	Specific to a message (%ASA-4-106100) that appears frequently; denotes the number of times the flow was permitted or denied by the ACL entry in the configured time interval.
hit_cnt_interval	Specific to a message (%ASA-4-106100) that appears frequently; denotes the interval in which the hit count is accumulated.
protocol	Protocol (TCP/UDP, etc.).
src_host	The source host, either an IP address or string, denoting the hostname.
src_interface	The source interface.
src_post	The source port.

To apply the Cisco ASA parser to your search:

1. In the search query field, type any keyword expression to filter on your IIS logs (or type an asterisk [*] wildcard).
2. Type a pipe symbol (|) and parse using **public/cisco/asa**.
3. Extend your query using operators such as **where** or **sort** with the named fields (listed above).

For a list of suggested Cisco ASA searches, see [Searches for the Cisco ASA Parser](#).

Parsing Microsoft IIS Logs

The Microsoft IIS parser extracts and labels the following fields.



The Microsoft IIS parser assumes that logs are provided in the default IIS W3C format for IIS 7.0.

Field Name	Description
c_ip	The IP address of the client making a request.
cs_method	A requested action (for example, a GET method).
cs_uri_query	The query (if any) that a client was trying to perform. A URI query is necessary only for dynamic pages.
cs_uri_stem	The Universal Resource Identifier (URI), or target, of an action.
cs_user_agent	The browser type that a client used.
cs_username	The name of the authenticated user who accessed a server. Anonymous users are indicated by a hyphen (-).
s_ip	The IP address of a server on which a log file entry is generated.
s_port	The server port number configured for a service.
sc_status	The HTTP or FTP status code.
sc_substatus	The HTTP or FTP substatus code.

Field Name	Description
sc_win32_status	The Windows status code.
time_taken	The length of time that the action took (in milliseconds).

To apply the IIS Log parser to your search:

1. In the search query field, type any keyword expression to filter on your IIS logs (or type an asterisk [*] wildcard).
2. Type a pipe symbol (|) and parse using **public/iis**.
3. Extend your query using operators such as **where** or **sort** with the named fields (listed above).

For a list of suggested IIS searches, see [Searches for the Microsoft IIS Parser](#).

Windows 2008 Parser

The Windows 2008 parser parses out common fields present in the Windows 2008 application, security, and system logs. The parser works only with events collected via a remote Windows event log source.

The Windows 2008 parser extracts and parses out the following fields:

Field Name	Description
category	Identifies the log message category.
comp_name	The name of the computer where the log message originated.
dest_domain	The destination domain of an action identified by a log message.
dest_user	The target user impacted by the action identified in the log message (for example, a new user created by an administrator).
event_id	The Windows event identifier that corresponds to the type of action that generated a log message.
event_source	Name of the software component that logs a Windows event.
event_type	Identifies the Windows event type in the log message.
fail_reason	The failure reason of the action being reported.
group_name	The Windows security group name identified by the log message.
group_domain	The domain of the Windows security group identified by the log message.
logon_process	The logon process.
logon_type	A number identifies how a user logged on (such as Interactive, Network, RemoteInteractive, etc.). More information on what each number means can be found in Microsoft documentation.
msg_summary	Summarizes the log message.
msg_type	Identifies the message type (such as Audit Success, Audit Failure, Information, and so on).
preauth_type	The preauthentication type of the log message.
process_id	The process ID.

Field Name	Description
process_name	The name of a process involved.
rec_num	The Windows event record number of the log message.
result_code	Captures either result, error, or failure code of the action reported by the log message.
src_domain	The source domain of the action identified by the log message.
src_ip	The source IP from which the action identified by the log message originated.
src_port	The source TCP port of a request identified by the log message.
src_user	The user performing the action identified by the log message.
thread_id	The thread ID.
update_result	The result of the update being reported (applicable to update logs only).
wkstation	The workstation identified by the log message.

Apply the Windows 2008 parser to a search

To apply the Windows 2008 parser to your search:

1. In the search query field, type any keyword expression to filter on your Windows 2008 logs, or type an asterisk (*) wildcard.
2. Type a pipe symbol (|) and **parse using public/windows/2008**.
3. Extend your query using operators such as **where** or **sort** with the named fields (listed above).

For example:

* | **parse using public/windows/2008** | **where...**

For more examples, see [Searches for Windows 2008 Events](#).

Windows 2008 Update Log parser

The Windows 2008 Update Log parser parses common fields in the Windows Update log file. On Windows 2008 systems, the file is commonly located at: **C:\Windows\Windowsupdate.log**.

The Windows 2008 Update Log parser extracts the following fields:

Field Name	Description
component	The logging component message. This field identifies the update log operation.
error_code	The error code, if present.
kbnun	The knowledge base number associated with a logged event, if present.
update_result	The result of an update, if present.

Apply the Windows 2008 Update Log parser to a search

To apply the Windows 2008 Update Log parser to your search:

1. In the search query field, type any keyword expression to filter your Windows 2008 logs, or type an asterisk (*) wildcard.
2. Type a pipe symbol (|) and **parse using public/windows/2008**.
3. Extend your query using operators such as **where** or **sort** with the named fields (listed above).

Parsing XML

The **XML** operator uses a subset of the XPath 1.0 specification to provide a way for you to parse fields from XML documents. Using it, you can specify what to extract from an XML document using an XPath reference.



Ingested XML files must be well-formed and valid in order to be parsed by the XML operator. If the XML is not valid, you will receive an error.

Syntax:

- ... parse XML [field=X] xpath_expressions* [as field_names*] [nodrop]

Rules:

- If no **field** is specified, then **_raw** is used.
- If the XPath is not valid, an error is thrown.
- If the number of field names don't match the specified XPath, an error is thrown.
- If the field is not well-formed XML, **null** is returned, unless you have specified **nodrop**.
- If the XPath doesn't match anything in the document, then **null** is returned, unless you have specified **nodrop**.
- If the XPath matches an element, then its string representation is returned.
- If the XPath matches multiple elements, then the first one is returned.

Examples:

Extract a field from an XML document.

For example, from this document:

```
<af type="nursery" id="102" timestamp="Nov 20 04:41:11 2013"
intervalms="1089510.533">
<minimum requested_bytes="48" />
<time exclusiveaccessms="0.163" meanexclusiveaccessms="0.163" threads="0"
lastthreadtid="0x0000000034520C00" />
<refs soft="40652" weak="35055" phantom="594"
dynamicSoftReferenceThreshold="10" maxSoftReferenceThreshold="32" />
<nursery freebytes="0" totalbytes="324978688" percent="0" />
<tenured freebytes="61087704" totalbytes="553484288" percent="11" >
<soa freebytes="33414104" totalbytes="525810688" percent="6" />
<loa freebytes="27673600" totalbytes="27673600" percent="100" />
<refs soft="40619" weak="29867" phantom="586"
dynamicSoftReferenceThreshold="10" maxSoftReferenceThreshold="32" />
```

```
<time totalms="91.622" />
</af>
```

you can extract information using an XPath reference, such as:

```
* | parse XML "/af/@type"
```

This will add the value of the attribute `type`, of the root `af` element, in a field called `/af/@type`, with value `nursery`.

The results are:

#	Time	/af/@type	Message
1	03/24/2014 15:45:42.735	nursery	<af type="nursery" id="97" ti <minimum requested_bytes="528 <time exclusiveaccessms="0.16 ... Host: 12.177.21.34 ▼ Name: Http Input
2	03/24/2014 15:45:32.686	nursery	<af type="nursery" id="102" t <minimum requested_bytes="48"

Extract a more complex field.

Use a query such as:

```
* | parse xml "/af/minimum/@requested_bytes"
```

This will add the value of the attribute `requested_bytes`, of the root `af/minimum` element, in a field called `/af/minimum@requested_bytes`, with the number of requested bytes.

#	Time	/af/minimum/@requ...	Message
1	03/24/2014 15:45:42.735	528	<af type="nursery" id="97" timestamp= <minimum requested_bytes="528" /> <time exclusiveaccessms="0.162" meane ... Host: 12.177.21.34 ▼ Name: Http Input ▼ Catego
2	03/24/2014 15:45:32.686	48	<af type="nursery" id="102" timestamp <minimum requested_bytes="48" />

XPath subset limitations

The full XPath 1.0 specification is not supported. In order to increase performance, Sumo Logic supports a subset of the specification, including the following caveats:

Forward only:

The XML operator only allows XML paths to go deeper into the tree. For example, this expression is not allowed:

- `/af/nursery/../../@type`

Full location paths

You must specify the full path to the elements you want to extract. This means that "self-or-descendant" expressions are not supported. For example, the following paths are not allowed:

- `//af`
- `/af//nursery`

No expanded syntax axis specifiers

Expanded syntax is not supported. For example, the following expressions cannot be used:

- `/child::af`
- `/descendant-or-self::af`

Group Operator and Group-By Functions

Aggregating functions evaluate messages and place them into groups. The **group** operator is used in conjunction with group-by functions. When using any grouping function, the word **by** is sufficient for representing the group operator. The typical construction when using group-by functions is:

grouping_function by (fieldname)

Aggregating (group-by) functions include:

- count
- count_distinct
- sum
- avg
- stddev
- first
- last
- min
- max
- pct



The **withtime**, **most_recent**, and **least_recent** operators are not considered standalone operators; they are only used with the first and last operators to enable those operators to be used in Monitors.



By default, the ordering is not defined inside of groups created using a group-by expression. To order your results, use the sort operator.

Group Operator

Syntax:

- ... | group-by_function (field_to_operate_on) group by (field_to_group_by)
- **Tip:** You can use **by** instead of **group by** so **count group by user** is equivalent to **count by user**.

Rules:

- Cannot be used with the **summarize** operator.
- When parsing and naming (aliasing) fields, avoid using the names of grouping functions or other operators as field names.
- When using **count**, or any grouping function, remember to include the underscore before the field name (sort by _count).

Examples:

- * | parse "GET * " as url | count by url | sort by _count | limit 10
- status AND down | parse regex "user=(?<user>.*)" | parse regex "host=(?<msg_host>.*)" | count

by user

- `_sourceCategory=apache | parse "*" as src_ip | parse "GET *" as url | count by src_ip | sort by _count`



All Sumo Logic system-generated fields begin with an underscore (`"_"`). Group-by functions always create a Sumo Logic field named with a combination of an underscore (`"_"`) and the function name. Using the function `count` inserts a field into the pipeline called `_count`. The function `count_distinct` inserts a field into the pipeline called `_count_distinct`.

avg

The averaging function (avg) calculates the average value of the numerical field being evaluated within the time range analyzed.

Syntax:

- `avg(numerical_field)`

Rules:

- Creates field named `_avg`

Example:

`... | avg(request_received) group by hour`

Sample log message:

```
Aug 2 04:06:08 : host=10.1.1.124: local/ssl2 notice mcpd[3772]:  
filesize=20454: diskutilization=0.4 : 01070638:5: Pool member  
172.31.51.22:0 monitor status down.
```

Example based on sample log message above:

`disk* | parse "diskutilization=*" as disk | avg(disk) group by _sourceCategory | sort by _avg`

This query finds all messages that contain the term `disk*` and parses out all that have a `diskutilization= value`. It then extracts the value of diskutilization into field `disk`. The next statement finds the average disk utilization by category. Effectively, it gives you a picture of how your hosts are doing on average based on categorization of log sources you've chosen.

count, count_distinct, and count_frequent

Aggregating (group-by) functions are used in conjunction with the group operator and a field name. Only the word `by` is required to represent the group operator. The count function is also an operator in its own right and therefore can be used with or without the word `by`.

count

Counts total number of messages that match the keyword search within the time range analyzed.

Syntax:

- count
- count by

Rules:

- Creates field named `_count`

Examples:

- `... | count as countOfPort group by srcAddress, tgtAddress`
- `* | parse "GET * " as url | count by url | sort by _count`

count_distinct

Counts only distinct occurrences of the value of a field being counted within the time range analyzed.

Syntax:

- `count_distinct(fieldname)`

Rules:

- Creates field named `_count_distinct`

Examples:

- `... | count_distinct(username) group by hostname`
- `_sourceCategory=*apache* | parse "*" -" as src_ip | count_distinct(src_ip)`



By default, ordering is not defined inside of groups created using a group-by expression. To order your results, use the sort operator.

If the number of distinct items returned is less than 10,000, the `count_distinct` function provides an exact number. If the number of distinct items returned is larger than 10,000, `count_distinct` instead uses an approximate algorithm, and displays a message that explains, "count_distinct saw more than 10000 values, results may be approximate."

The approximation algorithm uses a relative error parameter of 2%, for example:

- 65% of the time, results are within +/- 2%.
- 95% of the time, results are within +/- 4%.
- 99% of the time, results are within +/- 6%.

So for example, if the true count of distinct items is 1,000, the result returned by the approximation algorithm is between 950 and 1050 about 95% of the time.

The error parameter value is important to making the `count_distinct` function return results quickly and in a scalable way.

count_frequent

The `count_frequent` function can be used in cases where you want to identify the most common values for aggregations with over 10,000 distinct groups. This query returns the highest-count 10,000 results in sorted order. The resulting count field is called `_approxcount` because it is only an *estimate* of the true count; the estimate may be incorrect, but can only be over (it will never be under).

The `count_frequent` function is followed immediately by one or more field names.

Syntax:

- `count_frequent fieldname`
- For multiple fields: `count_frequent field1, field2, field3 (and so on)`

Rules:

- Creates field named `_approxcount`

Example:

- `* | parse "srcIP=*, url=*" as srcIP, url | count_frequent srcIP, url`

The limitations for using this operator are:

- Cannot be used with other aggregating functions like `sum` or `avg`.
- Sort is built into the query and defaults to a most-to-least order.

first and last

In general terms, **first** finds the earliest occurrence in search results, and **last** finds the result that follows all others, based on the sort order for the query

first

The default sort order for returned messages is reverse chronological—most recent descending to oldest. So **first** finds the most recent value of the field being evaluated within the time range. However, if you have specified a sort order other than descending chronological, then **first** finds the message that precedes all others based on the sort order defined in your query.

If there is no sort order specified for returned results (for example, when using `limit 20`), then **first** simply returns the first result encountered without respect to date or list order.

Syntax:

- `first(fieldname)`

Rules:

- Creates field named `_first`

Example:

```
... | first(error_message) group by hostname
```

last

Finds the last value of the field being evaluated within the time range and according to the specified sort order. Remember that the default order for returned messages is reverse chronological—most recent descending to oldest. Therefore, **last** is the oldest result in the returned list. If you have specified an order other than reverse chronological, then **last** finds the ending message that follows all others based on your sort order.

Syntax:

- **last(fieldname)**

Rules:

- Creates field named **_last**

Example:

```
... | last(status_code) group by hostname
```

Sample log message:

```
Aug 2 04:06:08 : host=10.1.1.124: local/ssl2 notice mcpd[3772]:  
filesize=20454: diskutilization=0.4 : 01070638:5: Pool member  
172.31.51.22:0 monitor status down.
```

Example based on sample log message:

```
disk* | parse "diskutilization=" as disk | disk>0.8?1:0 as overcapacity | last(overcapacity) by _sourceHost |  
sort by _last
```

This query finds all messages that contain the term **disk*** and parses out all that have a **diskutilization=** value. It then extracts the value of diskutilization into field **disk**. It then determines if that value is greater than 80% and will find the last occurrence of that value per host effectively producing a list of hosts that have disk utilization that is over 80%.

Using first and last operators in Monitors

Data isn't always ingested in perfect order, meaning that log messages may arrive out of sequence. In one-off queries that use the first operator or last operator nonsequential logs aren't an issue. However, to save a query as a Monitor, because the query is run continuously, logs need to be in perfectly sequenced to produce results. To make sure that they are in perfect order, you'll need to explicitly tag fields with the **withtime** operator.

Using **withtime** forces log messages to be put in perfect order, which then allows you to add queries that contain the first or last operator. It created a field named **xxx_withtime** that will appear as part of your search results. The **most_recent** and **least_recent** operators allow you to order data from newest to oldest.



The **withtime** operator is not considered a standalone operator; it's only used with the first and last operators to enable those operators to be used in Monitors.

Syntax:

- * | parse ... as status | withtime status | most_recent(status_withtime) by _sourcehost
- * | parse ... as status | withtime status | least_recent(status_withtime) by _sourcehost

Examples:

Find the most recent visitors to our site by IP. Say we'd like to keep an eye on visitors that hit our site from different countries. Running a query like:

ip* OR *address

| parse regex "(?<IP>\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"

| lookup latitude, longitude, country_code from geo://default on ip=IP

| where !isNull(country_code)

| withtime IP

| most_recent(ip_withtime) by country_code

produces results like:

Messages		
Aggregates		
Page: 1 of 2		
#	country_code	_mostrecent
1	JP	54.24.122.9
2	DK	195.18.122.131
3	DE	83.23.178.166
4	GB	213.24.191.2
5	PH	170.25.160.21
6	SI	89.14.12.238
7	SG	54.25.96.238
8	SE	79.13.117.166
9	CR	18.32.3.146
10	FR	78.31.5.25
11	IT	95.14.32.46
12	CO	190.25.81.92
13	CN	16.3.42.19
14	CL	200.14.10.59
15	RU	37.20.9.248
16	IN	14.41.84.194
17	CH	194.88.19.32
18	IL	80.179.69.21
19	FI	194.100.73.10
20	US	19.143.33.1
21	CA	174.90.12.5
22	IE	54.246.21.68
23	ID	116.24.103.37
24	BV	209.18.126.35
25	BR	201.49.208.18

Find the most recent user logged in by account. If you'd like to find the user who most recently logged in (per account or organization, which we've used in our example) you can use a first operator, in a query that looks something like:

```
...| parse "Successful login for user **", organization: "" as user,org_id | withtime user | first(user_withtime) by org_id
```

This query would return a list of organizations and a list of the user ID that most recently logged in. You could save the query as a Monitor to keep a constant eye on logins.

min and max

Use the min and max functions to find the smallest or largest value in a set of values.

max

Extracts the maximum value of the numerical field being evaluated within the time range.

Syntax:

- max(numerical_field)

Rules:

- Creates field named `_max`

Example:

```
... | max(request_received) group by hour
```

min

Extracts the minimum value of the numerical field being evaluated within the time range.

Syntax:

- min(numerical_field)

Rules:

- Creates field named `_min`

Example:

```
... | min(request_received) group by hour
```

most_recent and least_recent

The `most_recent` and `least_recent` operators, used with the `withtime` operator, allow you to order data from newest to oldest.



The `most_recent`, and `least_recent` operators are not considered standalone operators; they are only used with the `withtime` operator in queries that use the first and last operators to enable those queries to be saved as Monitors.

Using `withtime` forces log messages to be put in perfect order, which then allows you to add queries that contain the first or last operator. It created a field named `xxx_withtime` that will appear as part of your search results. The `most_recent` and `least_recent` operators allow you to order data from newest to oldest.

Syntax:

- `* | parse ... as status | withtime status | most_recent(status_withtime) by _sourcehost`
- `* | parse ... as status | withtime status | least_recent(status_withtime) by _sourcehost`

Examples:

Find the most recent visitors to our site by IP. Say we'd like to keep an eye on visitors that hit our site from different countries. Running a query like:

ip* OR *address

```
| parse regex "(?<IP>\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"  
| lookup latitude, longitude, country_code from geo://default on ip=IP  
| where !isNull(country_code)  
| withtime IP  
| most_recent(ip_withtime) by country_code
```

produces results like:

Messages		Aggregates	
<div> <div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> </div> </div>		Page: 1	of 2
#	country_code	_mostrecent	
1	JP	54.24.122.9	
2	DK	195.18.122.131	
3	DE	83.23.178.166	
4	GB	213.24.191.2	
5	PH	170.25.160.21	
6	SI	89.14.12.238	
7	SG	54.25.96.238	
8	SE	79.13.117.166	
9	CR	18.32.3.146	
10	FR	78.31.5.25	
11	IT	95.14.32.46	
12	CO	190.25.81.92	
13	CN	16.3.42.19	
14	CL	200.14.10.59	
15	RU	37.20.9.248	
16	IN	14.41.84.194	
17	CH	194.88.19.32	
18	IL	80.179.69.21	
19	FI	194.100.73.10	
20	US	19.143.33.1	
21	CA	174.90.12.5	
22	IE	54.246.21.68	
23	ID	116.24.103.37	
24	BV	209.18.126.35	
25	BR	201.49.208.18	

pct

The percentile function (pct) finds the percentile of a given field. Multiple pct functions can be included in one query.



The pct function will return an approximate result for searches that produce large volumes of data.

Syntax:

- ...| pct(fieldname [, percentile]) [as alias] [by group]

Rules:

- Creates a field named: `_fieldname_pct_percentile`

Example:

*** | parse "data=" as data | pct(data, 95)**

Sample log message:

```
Aug 2 04:06:08 : host=10.1.1.124: local/ssl2 notice mcpd[3772]:
filesize=20454: diskutilization=0.4 : 01070638:5: Pool member
172.31.51.22:0 monitor status down.
```

Example based on sample log message:

file* | parse "filesize=" as filesize | pct(filesize, 75), pct(filesize, 95) by _sourceHost

Running this query creates fields named `_filesize_pct_75` and `_filesize_pct_95`.

stddev

The standard deviation function (stddev) finds the standard deviation value for a distribution of numerical values within the time range analyzed and associated with a group designated by the "group by" field.

Syntax:

- `stddev(numerical_field)`

Rules:

- Creates field named `_stddev`

Example:

... | stddev(request_received) group by hour | sort by _stddev

sum

Sum adds the values of the numerical field being evaluated within the time range analyzed.

Syntax:

- `sum(numerical_field)`

Rules:

- Creates field named `_sum`

Example:

... | sum(bytes_received) group by hostname

Sample log message:

```
Aug 2 04:06:08 : host=10.1.1.124: local/ssl2 notice mcpd[3772]:
filesize=20454: diskutilization=0.4 : 01070638:5: Pool member
172.31.51.22:0 monitor status down.
```

Example based on sample log message above:

```
file* | parse "filesize=*" as filesize | sum (filesize) group _sourceHost
```

Finds all messages that contain term **file*** and parses out all that have a **filesize=value**. It will then extract the value of filesize and will add all those values per host where those log messages are generated.

CHAPTER A

Sumo Logic Operators and Expressions

This section provides detailed syntax, rules, and examples for Sumo Logic Operators, Expressions, and Search Language.



Occasionally, new search operators and functions are introduced as Beta features. If you are interested in trying out new Beta operators for search, visit Sumo Logic Labs at [the Sumo Logic Support site](#). You must be logged in to the support site to visit Sumo Logic Labs.

Search Syntax Overview

The Sumo Logic Search Language operates on your entire log repository, no matter how many different log sources you have—in real time. The search query language is intuitive and efficient, allowing you to search terabytes of data and see results in seconds.

Query Syntax

The basis of Sumo Logic Search is a funnel or "pipeline" concept: beginning from all of your current Sumo Logic data, you enter keywords and operators separated by pipes ("|"). Each operator acts on the results from the previous operator to further process your results. Results are returned incrementally with the most recent messages displaying first. Additional messages are added progressively to the Messages tab as the search walks backward in time through all of your log data.

The syntax for a typical search query looks something like this:

keyword expression | operator 1 | operator 2 | operator 3

Keyword Expression: For simplicity, we refer to the first term in a search query as a "keyword" expression. In fact, this portion of the query is a very powerful full-text, Boolean search expression. The keyword expression also encompasses metadata searches for fields such as `_sourceCategory`, `_sourceHost`. For more on full-text search in queries, see [Keyword Search Expressions](#).

Operators: After filtering with an initial full-text search, the operators that follow can extract strings, parse known message components into fields, refine results using conditional expressions, and then group, count, or sort results. In addition, the **summarize** operator can be used to reveal patterns in a set of logs by automatically grouping messages with similar structures and common repeated text strings into clusters.

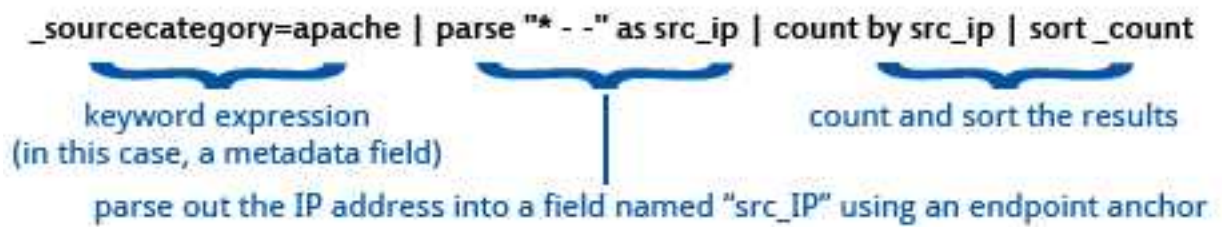
Pipe "|" Delimiter

The pipe delimiter is used to separate the keyword expression and each subsequent operator. Each pipe-delimited operator further processes search results from the preceding operator. You can string some operators in a series within a single pipe (like **parse** and **where**), but if you are not sure of the syntax, always add the pipe.

Syntax:

- Follow keyword search expression with a pipe "|"
- Precede each operator with a pipe "|"

Example:



User-Created Fields

You can parse or extract values and assign a user-created (aliased) field name to the result. The field is valid only for the current search, and does not carry over to new searches. When creating aliased fields, there are a few rules that apply:

- Field names can contain alphanumeric characters, hyphens, and underscores, but should always start and end with an alphanumeric character. Sumo Logic fields always begin with an underscore, such as `_sourceCategory`, `_sourceHost`, or `_count_distinct`. Here are two examples of queries that generate a user-created field called `src_IP`:
 - `* | parse regex "(?<src_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"`
 - `_sourceCategory=apache | parse "*" as src_IP`
- Multiple fields can be extracted and named within a single query. For example, the query below creates fields "type" and "user":
 - `_sourcehost=vpn3000 | parse "Group [*] User [*]" as (type, user) | count type | sort by _count`
- Aggregating functions also automatically generate a field. Using the count operator creates a field called `_count`. The sum operator creates a field called `_sum`. The max operator creates a field called `_max`, and so forth.
- User-created fields should not be named with reserved words such as the names of Sumo Logic operators like `group` or `sum`.

For information on parsing fields, see [Extracting and Naming a Field](#)

Refining a Query

Often there are many ways to write a query. Sometimes it is an iterative process to refine your query to get the search results that you are looking for.

Say you are looking to parse out a search query from your message logs. In this example, you want to find the query used in the event "save view operator: Done publishing."

You may try this query first, but it will NOT work:

```
"save view operator: Done publishing" | parse "searchQuery=| save view" as query | where query matches "(count|summarize)*"
```

But using the `parse regex` operator, this query does work:

```

"save view operator: Done publishing"
| parse regex ".*searchQuery=(?<qqq>.(?:count|summarize).*)?"
| save view"
| where qqq != "" and ! isNull(qqq)

```

Providing results like this:

#	Time	qqq	Message
1	03/27/2014 08:31:26.694	(_sourceCategory=forge or _sourceCategory=receiver or _sourceCategory=cloudcollector) "Message stats,	2014-03-27 08:31:26 [logger=stream_pipe 78AC0928657812DF-1]

Instead of using parse regex, you can use the matches operator, like this:

```

"save view operator: Done publishing"
| parse "searchQuery=*" | save view" as qqq
| if (qqq matches "*count*", "COUNT", "") as hasCount
| if (qqq matches "*summarize*", "SUMMARY", "") as hasSummary
| where hasCount != "" or hasSummary != ""

```

which gives you these results:

#	Time	qqq	hascount	hassummary	Message
1	03/27/2014 08:32:34.745	I parse "remote_ip=*" as remote_ip lookup latitude, longitude, country_code, country_name, region, city, postal_code, area_code, metro_code from geo://default on ip = remote_ip	COUNT		2014-03-27 08 [localUserNam [thread=MTP-F

Or this query will work as well:

```

"save view operator: Done publishing"
| parse "searchQuery=*"
| save view" as qqq
| where (qqq matches "*count*") or (qqq matches "*summarize*")

```

to provide results like this:

#	Time	qqq	Message
1	03/27/2014 08:33:30.262	I parse "remote_ip=*" as remote_ip lookup latitude, longitude, country_code, country_name, region, city, postal_code, area_code, metro_code from geo://default on ip = remote_ip	2014-03-27 08:33:30 [logger=stream_pipe A4FE87340BCEF0A2-1] [authCustomer:0000

See also:

- [Parse Regex](#)
- [Matches Operator](#)

Conditional Expressions

Conditional expressions filter results by evaluating against a criterion. Only results that meet the criterion are returned.

You can use the **where** operator:

- With standard operators (like <, >, =>, =, <>).
- For matching quoted string literals (where x = "disk value", or where x matches "someth?ng").
- For matching against multiple values using "in" (where x in (a, b, c)).
- For values that do not match using "not in" (where x not in (d, e, f)).

Ternary expressions are a shorthand form of if-else statement. A ternary expression sets up a test that evaluates to only one of two values.

Tell me more about:

- [The **where** operator](#)
- [Ternary expressions](#)

Where Operator

To filter results in a search query, use "where" as a conditional operator. The where operator must appear as a separate operator distinct from other operators, delimited by the pipe symbol ("|"). In other words, the following construct will not work and will generate a syntax error:

This query will NOT work:

```
...| parse "seconds=*" as time where > 5
```

Instead, separate the **where** operator from the preceding **parse** operator like this:

```
...| parse "seconds=*" as time | where time > 5
```

Syntax:

- ... | where <Boolean expression> | ...

Rules:

- The pipe delimiter is required to separate the **where** operator as a distinct query operator.
- The **where** operator *cannot* be used inline as a query clause, like "... | extract a where b==something |...."
- * is allowed for string matching (zero or more than one character).
- Multiple where-expressions are processed in the order they are specified, with each subsequent **where** operator further filtering results.
- If you are using "in" or "not in" to match integers, cast "x" to a number first.

Examples:

- ... | where a<b | count by _sourceHost
- ... | where a=x
- ... | where a>=x

- ... | where a<=x
- ... | where a<x
- ... | where x="some string"
- ... | where x matches "some string"
- ... | where x matches "fail*"
- ... | where x<10
- ... | where user<>"root"
- ... | where error="fail*"
- ... | num(x) | where x in (4, 3, 5)
- ... | where x in ("error", "fail")
- ... | where x not in ("error", "fail")

Using the "not" option

If you need a query using the **where** operator, where xxx DOES NOT match yyy, use "!" followed by the **matches** operator enclosed in parenthesis.

For example:

```
...| where !(<field xxx> matches "<value yyy>") | ...
```

or:

```
...| where !(status matches "200")
```

Use where to check for null values

For details, see [isNull operator](#).

IF Operator

There are two forms of ternary expression you can use in Sumo Logic queries: one is constructed using the **IF** operator, and the other uses the question mark (?) operator. The syntax varies slightly, but the results are equivalent. You can use the syntax you're most comfortable with.

These expressions are used to evaluate a condition as either true or false, with values assigned for each outcome. It is a shorthand way to express an if-else condition. On the basis of the test, the entire expression returns value_if_true if the condition is true, else value_if_false if the condition is false. The two sub-expressions (value_if_true and value_if_false) must have the same type.

Using the IF operator

Syntax:

- ... | if(condition, value_if_true, value_if_false) as alias_field | ...

Examples:

- ... | if(status_code matches "5*", 1, 0) as servererror | ...
- ... | if(status_code matches "2*", 1, 0) as success | ...

Using the Question Mark (?) operator

Syntax:

- ... | alias_field = condition ? value_if_true : value_if_false | ...

Examples:

- disk_usage > threshold ? "disk full" : "OK" as status
- a < b ? a : b // This is the same as min(a, b)



For information on handling null values, see [isNull operator](#).

Manually Casting String Data to a Number

Most data in Sumo Logic is stored as string type data. Metadata fields are stored as string data. And parsed fields are, by default, parsed as string type data. Sumo Logic will implicitly cast string data to a number type assuming it is clear that you need a number to perform an action, such as a math calculation or when using a function like **sum** or **avg**. However, if there is any ambiguity about whether a number is required, the data remains string data.

This detail can be important when you are building queries. There are at least two cases where you will need to manually cast string data to a number so that you get the results that you expect:

- When using the **where** operator to match integers like this:
 - **where value in** (integer_value1, integer_value2, integer_value3)
 - **where value not in** (integer_value1, integer_value2, integer_value3)
- When you need to numerically sort a series of results from a query, like in this example:
 - *** | parse "took *ms" as duration | toLong(duration) | sort by duration**

In the first case, if your statement looks something like "where some_value in (1, 2, 4, 16)" and you need to match (or not match) integers, then you will first need to cast "some_value" to a number.

In the second case, the results will sort out of order as text values if you do not first cast the field "duration" to a number. After the field is cast to a number type, the sort order will produce expected results.

Sumo Logic accepts these functions for casting string data to a number:

- num()
- number()
- toLong()

When casting a field to a number, remember to separate the casting statement with a pipe, like this:

```
* | parse "OSload *ms" as boot_time | number(boot_time) | sort by boot_time
```

Accum operator

The **accum** operator calculates the cumulative sum of a field. The **accum** operator can be used to find a count by a specific time interval, and can be used to find a total running count across all intervals.

Syntax:

- `accum field [as alias] [by field1, field2, ...]`

Rules:

- An alias for **accum** is optional. When an alias is not provided, **_accum** is the default alias.
- Specified fields must contain numeric values.
- If a row contains non-numeric values, that row will be skipped.
- To add a query that includes an accum operator to a Dashboard, you must add a group by function before the **accum** operator.

Examples:

Requests by running total. With the accum operator, we can find the number of requests by a user as a running total. Running a query similar to:

```
_sourceCategory=IIS (Wyatt OR Luke)
| parse using public/iis
| timeslice by 1m
| count as requests by _timeslice,cs_username
| sort by _timeslice asc,cs_username
| accum requests as running_total
```

produces results of a running total of all requests, similar to:

Messages		Aggregates			
		<div> <div>⏪</div> <div>⏩</div> <div>Page: 1 of 1</div> <div>⏪</div> <div>⏩</div> </div>			
#	Time	cs_username	requests	running_total	
1	01-25-2013 14:27:00	Wyatt	12	12	
2	01-25-2013 14:27:00	Luke	8	20	
3	01-25-2013 14:28:00	Wyatt	12	32	
4	01-25-2013 14:28:00	Luke	8	40	
5	01-25-2013 14:29:00	Wyatt	12	52	
6	01-25-2013 14:29:00	Luke	8	60	

Running total by user name. Another option is to find a running total for each user's requests. Running a query similar to:

```
_sourceCategory=IIS (Wyatt OR Luke)
| parse using public/iis
| timeslice by 1m
| count as requests by _timeslice,cs_username
| sort by _timeslice asc,cs_username
| accum requests as running_total by cs_username
```

produces results of a running total for each user's requests, similar to:

Messages		Aggregates		
<div> <div> <div>⏪</div> <div>⏩</div> </div> <div> <div>⏴</div> <div>⏵</div> </div> </div>		Page: 1	of 1	<div> <div>⏴</div> <div>⏵</div> </div>
#	Time	cs_username	requests	running_total
1	01-25-2013 14:28:00	Wyatt	4	4
2	01-25-2013 14:28:00	Luke	1	1
3	01-25-2013 14:29:00	Wyatt	12	16
4	01-25-2013 14:29:00	Luke	8	9

CIDR Operator

The CIDR operator allows you to leverage CIDR (Classless Inter-Domain Routing) notations to analyze IP network traffic in order to narrow analysis to specific subnets. CIDR notations specify the routing prefix of IP addresses ([learn more here](#)). Using the CIDR operator, you can determine the amount of traffic between network segments, review events from hosts within a specified network segment, or even use a `not` operator to find addresses that didn't originate from a particular network segment.

To use the CIDR operator, you must first parse IP addresses from messages, then match those IP addresses against a network segment via CIDR notations.

Syntax:

- **`getCIDRPrefix (String IP_address, int prefix_length) as xx`**

Using this syntax produces CIDR prefixes of previously parsed IP addresses.

- **`compareCIDRPrefix(String IP_address1, String IP_address2, int prefix_length) as xx`**

Using this syntax returns a Boolean. **`true`** indicates that prefixes are the same; **`false`** indicates that prefixes are different.

- **`maskFromCIDR(int prefix_length) as xx`**

Using this syntax returns an IP address subnet mask, similar to:

CIDR	Mask
32	255.255.255.255
31	255.255.255.254
30	255.255.255.252
29	255.255.255.248
...	...

Rules:

- prefix is an integer from 0-32.
- IP_address is an IPV4- or IPV6-formatted string.

Examples:

Review events from a specific network segment:

1. Search for the events. For example, let's say we'd like to review firewall logs:
`(denied OR rejected AND _sourcecategory=firewall | ...`
2. Parse the IP addresses. For example:
`... | parse "ip=*, " as ip_address ...`
3. Compare to the full CIDR notation you requested. For example, 10.10.1.32/27:
`... | where compareCIDRPrefix("10.10.1.32", ip_address, toInt(27)) | ...`
4. Keep matching records, and drop non-matching records from search results:
`... | count by ip_address`

Review events not from a specific network segment:

1. Search for the events. For example, let's say we'd like to review firewall logs:
`(denied OR rejected AND _sourcecategory=firewall | ...`
2. Parse the IP addresses. For example:
`... | parse "ip=*, " as ip_address ...`
3. Compare to the full CIDR notation you requested, and drop matching records. For example, 10.10.1.32/27:
`... | where !compareCIDRPrefix("10.10.1.32", ip_address, toInt(27)) | ...`

Concat Operator

The Concat operator allows you to concatenate or join multiple strings, numbers, and fields into a single user-defined field. It concatenates strings end-to-end and joins them into a new string that you define. For example, to concatenate the words "foot" and "ball" would give you "football". You can also use punctuation and spaces in quotes to concatenate strings in a readable way.

In another example, let's say a log message has a table with the elements of a mailing address, but separated into different fields such as `Street_Number`, `City`, `State`, and `Zip_Code`. You can use the concatenate operator to assemble the fields into a new field called `Mailing_Address` for a customer.

In another example, if you had a log message of an incident with four fields, such as `Signature_Name`, `Vendor_Signature`, `Incident_Detail_URL`, and `Analyst_Assessment` that you wanted to combine into a single field (a single string) called `Event_Detail`, the concatenate operator would also allow you to do this.

Syntax:

- `concat(field1, field2) as [fieldname]`

Rules:

- You must define a name for the new `[fieldname]` to concatenate the named fields. There is no default.
- You can use punctuation and spaces in quotes to concatenate strings in a readable way.
- A null field is treated as empty string.
- The operator allows 2 to 16 inputs. To use more than 16 inputs, you can combine operators. See example.
- AND and OR are not supported.

Examples:

Concatenate fields with and without punctuation.

Let's say you had the following fields: field1 = time, field2 = 4, field3 = logs. Using this query:

```
concat (field1, field2, field3) as new_string
```

would return: new_string = time4logs

If you add punctuation and spaces in quotes, like this:

```
concat (field1, " ", field2, " ", field3) as new_string
```

you would get: new_string = time 4 logs

Concatenate fields to create an IP Address.

In this example, to create an IP address out of separate message log fields, concatenate four number fields with punctuation to complete a new field named **ip_address**.

```
... | concat(octet1, ".", octet2, ".", octet3, ".", octet4) as ip_address
```

Concatenate first and last names.

In this example, you could concatenate fields for a first and last name to create a new field called **fullName**.

```
... | concat(firstName, " ", lastName) as fullName
```

Formatting dates.

You can use the Concat operator to format dates, as shown:

```
... | concat (month,"/", day,"/",year) as date
```

Concatenate more than 16 inputs.

To use more than 16 inputs with the concat operator, you can combine operators, using one of the following formats:

- **concat(field1, field2, ...) as b | concat(b, field17, field18,...) as c | ...**
- **concat(concat(field1, field2, ...), field17, field18,...)**

See Also:

For information on formatting strings, see [Format Operator](#).

CSV Operator

The CSV operator allows you to parse CSV (Comma Separated Values) formatted log entries. It uses a comma as the default delimiter.

For example, let's say you have a .csv file that maps internal IP addresses to your data center locations and names. Once the .csv file is ingested into Sumo Logic, you can use the CSV operator to parse the fields of the file and populate a lookup table. Then you could use the Geo Lookup operator to map your data center IP addresses and display them on a map of the world.



To parse delimited log entries other than CSV files, such as space delimited files, use the [Split operator](#).

Syntax:

Extract fields using index:

- csv fieldName extract 1 as A, 2 as B, 5 as E, 6 as F

Extract fields using position:

- csv fieldName extract A, B, __, __, E, F

Extract from an existing field:

- parse "start*end" as fieldName | csv fieldName extract 1 as A, 2 as B, 5 as E, 6 as F

Specify an escape, and quote character:

- csv fieldName escape="\", quote="\" extract A, B, __, __, E, F

Rules:

- By default, the CSV operator uses a comma (,) for a delimiter, backslash (\) for an escape character, and (") quote for a quote character.
- A field name is always required.

Examples:

Parse comma delimited fields.

Use the following query to parse a CSV file's comma delimited fields as shown:

```
_sourceCategory=csv  
| csv _raw extract 1 as user2, 2 as id, 3 as name
```

which provides results like:

#	Time	user2	id	name	Message
1	05/08/2014 16:59:56.761	user10	10	Franklin Trevaleon	user10,10,"Franklin Trevaleon" Host: 108.248.126.124 ▼ Name: Http Input ▼ Category: csv ▼
2	05/08/2014 16:59:56.761	user9	9	Napoleon Trois	user9,9,"Napoleon Trois" Host: 108.248.126.124 ▼ Name: Http Input ▼ Category: csv ▼

Parse a stream query and extract search terms.

"Starting stream query" | parse "query=[*], queryId" as query | csv query extract searchTerms, op1, op2, op3

This produces results like:

#	Time	query	searchterms	op1	op2	op3	Message
121	04/21/2014 15:28:40.099	* sample processingRate=10000 timeout=600000 assign_signatures infer_signatures count by signature	*	sample processingRate=10000 timeout=600000	assign_signatures	infer_signatures	2014-04-21 15: [module=STREAN [logger=stream 850080749-66]

For details on using CSV files, see [Structuring CSV Files](#).

For more information on parsing CSV files, see [Lookup Operator](#) and [Save Operator](#).

Structuring CSV Files

Sumo Logic supports CSV files with the following restrictions:

- The CSV file must contain a header line.
- All values in the CSV file need to be wrapped in quotes.
- No spaces are allowed between quotes and values. For example:

```
"id","name","time"
```

```
"1","foo","6-15-12"
```

```
"2","zoo","6-14-12"
```

```
"3","woo","6-13-12"
```

Difference (Diff) Operator

The `diff` operator calculates the rate of change in a field between consecutive rows. To produce results, `diff` requires that a specified field contain numeric data; any non-numerical values are removed from the search results.

Diff doesn't sort data, but instead operates on rows in the order that they appear in the input stream, subtracting the number in a field from the number in the same field in the previous line.



The first line of results will never display diff results.

Adding a group by function to a **diff** operator query calculates the difference between consecutive values in each group. (Data from each group are calculated separately.) Grouping doesn't affect the order in which rows appear in the output stream.

Syntax:

- ...diff field [as alias] [by field1, field2, ...]

Rules:

- An alias for **diff** is optional. When an alias is not provided, **_diff** is the default alias.
- Specified fields must contain numeric values.
- If a row contains non-numeric values, that row will be skipped; **diff** uses the row before that (until it finds an acceptable row with a numeric value).
- The **diff** corresponding to the first row in any results is null (empty).
- To add a query that includes a **diff** operator to a Dashboard, you must add a group by function before the **diff** operator.

Examples:

Using diff to calculate the difference of a quantity between time points. Using diff with timeslice, you can run a query similar to:

```
* | parse "bytes transmited: *" as bytes | timeslice 1m | sum(bytes) as bytes by _timeslice |  
diff bytes as diff_bytes
```

to produce results similar to:

Whitelist

For whitelist mode, only fields you specify for inclusion are kept in the search output. For example, to strip out every field except for method and status_code, your query would be:

```
_sourceCategory=access_logs | parse using public/apache | fields method, status_code
```

The search results would look like this:

Messages				
Page: 1 of 4				
#	Time	method	status_code	Message
1	10/18/2012 12:44:20.000	GET	404	91.140.35.249 - - [18/Oct/2012:12:44:20 -0700] "GET /signup.php HTTP/1.1" 404 468 "-" "Mozilla/5.0 (Windows NT 5.1; rv:8.0) Gecko/20100101 Firefox/8.0" Host: www.freessoft.org Name: /var/www/vhosts/freessoft.org/statistics/logs/access_log Category: access_logs
2	10/18/2012 12:44:20.000	GET	404	91.140.35.249 - - [18/Oct/2012:12:44:20 -0700] "GET /signup.php HTTP/1.1" 404 468 "-" "Mozilla/5.0 (Windows NT 5.1; rv:8.0) Gecko/20100101 Firefox/8.0" Host: www.freessoft.org Name: /var/www/vhosts/freessoft.org/statistics/logs/access_log Category: access_logs
3	10/18/2012 12:44:18.000	GET	404	91.140.35.249 - - [18/Oct/2012:12:44:18 -0700] "GET /login.php HTTP/1.1" 404 467 "-" "Mozilla/5.0 (Windows NT 5.1; rv:8.0) Gecko/20100101 Firefox/8.0" Host: www.freessoft.org Name: /var/www/vhosts/freessoft.org/statistics/logs/access_log Category: access_logs
4	10/18/2012 12:44:18.000	GET	404	91.140.35.249 - - [18/Oct/2012:12:44:18 -0700] "GET /login.php HTTP/1.1" 404 467 "-" "Mozilla/5.0 (Windows NT 5.1; rv:8.0) Gecko/20100101 Firefox/8.0" Host: www.freessoft.org Name: /var/www/vhosts/freessoft.org/statistics/logs/access_log Category: access_logs
5	10/18/2012 12:44:17.000	GET	404	91.140.35.249 - - [18/Oct/2012:12:44:17 -0700] "GET /register/ HTTP/1.1" 404 467 "-" "Mozilla/5.0 (Windows NT 5.1; rv:8.0) Gecko/20100101 Firefox/8.0" Host: www.freessoft.org Name: /var/www/vhosts/freessoft.org/statistics/logs/access_log Category: access_logs



Whitelist queries allow all system internal fields (fields prefixed with an underscore "_") to pass.

Blacklist

For blacklist mode, all fields except for those you explicitly remove remain in the search output. Blacklist mode is indicated with a minus sign "-" in a query. For example, to only remove the log_level, module, and process_id fields, your query would be:

```
_sourceCategory=*apache* | parse using public/apache | fields - log_level, module, process_id
```

Blacklist queries will also remove internal fields (fields prefixed with an underscore "_") when specified. For example:

```
_sourceCategory=*apache* | parse using public/apache | count by size | fields - _count
```



Make sure that your query does not repeat or duplicate individual fields, or your search query will fail.

Format Operator

The **Format** operator allows you to format and combine data from fields in message logs—including numbers, strings, and dates—into a single user-defined string. This allows data in message logs, such as dates or currency amounts, to be formatted as human readable, when otherwise it would be hard to decipher.



The **Concat operator** is a simpler version of the **Format operator**, and may be used instead for simpler use cases.

Syntax:

- `format(formatSpecifier, field1, field2, ..., fieldn) as [fieldname]`
- The Sumo Logic Format operator supports all Java `String.format` syntax, as defined in <http://docs.oracle.com/javase/7/docs/api/java/util/Formatter.html#syntax>

Rules:

- The first argument to the Format operator must be a format specifier, which is a string.
- You must define a name for the new [fieldname] to use Format. There is no default.
- The operator allows 2 to 16 inputs. To use more than 16 inputs, you can combine operators.
- AND and OR are not supported
- If a field is null or incompatible, an error will be thrown.
- Use the Format operator after the aggregate.

Examples:

Format two strings into one string.

In this query, we search for errors, then parse the field “fiveMinuteRate” as “rate”, then combine the text “Five Minute Rate is:” and the rate together as “formattedVal”.

```
error | parse "fiveMinuteRate=*" as rate | format("%s : %s", "Five Minute Rate is" , rate) as formattedVal
```

which results in:

#	Time	rate	formattedval	Message
1	04/17/2014 10:20:49.955	0.07	Five Minute Rate is : 0.07	2014-04-17 10:20:49.955 reporter-katta-1 Host: nite-katta-1 ▼
2	04/17/2014 10:20:49.430	0.07	Five Minute Rate is : 0.07	2014-04-17 10:20:49.430 reporter-katta-1 Host: nite-katta-1 ▼

Format numbers.

This query allows you to format number fields from a message log into a properly formatted, human readable currency amount.

```
format( "$%.2f",number) as currency
```

Formatting dates.

Use the following query to format fields in a message log into a readable date.

```
| parse "*" "*" "as year, month, day"  
| format ("%d/%d/%d", month, day, year) as date
```

Convert strings to uppercase.

Use this format specifier to convert strings to uppercase:

```
format("%S: %d", name, age) as personAge
```

For more options, see [toLowerCase](#) and [toUpperCase](#).

Mapping IP addresses with the geo lookup operator

Sumo Logic can match an extracted IP address to its geographical location on a map. To create the map, after parsing the IP addresses from log files, the **lookup** operator matches extracted IP addresses to the physical location where the addresses originated. Finally, geo-location fields are used by the Google Maps API to add the IPs to a map. The latitude and longitude fields are required; optional fields are country_code, country_name, region, city, postal_code, area_code, and metro_code. Depending on how specific you'd like the output to be, you can include all the optional fields, or choose a subset.

Syntax

To produce a map, your query should use the following syntax:

```
| parse "[ip_fieldname]" as [ip_address]  
| lookup latitude, longitude, [optional_geo_locator fields] from geo://default on ip=[ip_address]  
| count by latitude, longitude, [other_geo_locator fields]  
| sort_count
```

Because this syntax produces aggregate results, you can add a map to a Dashboard.

Example

Sample log message:

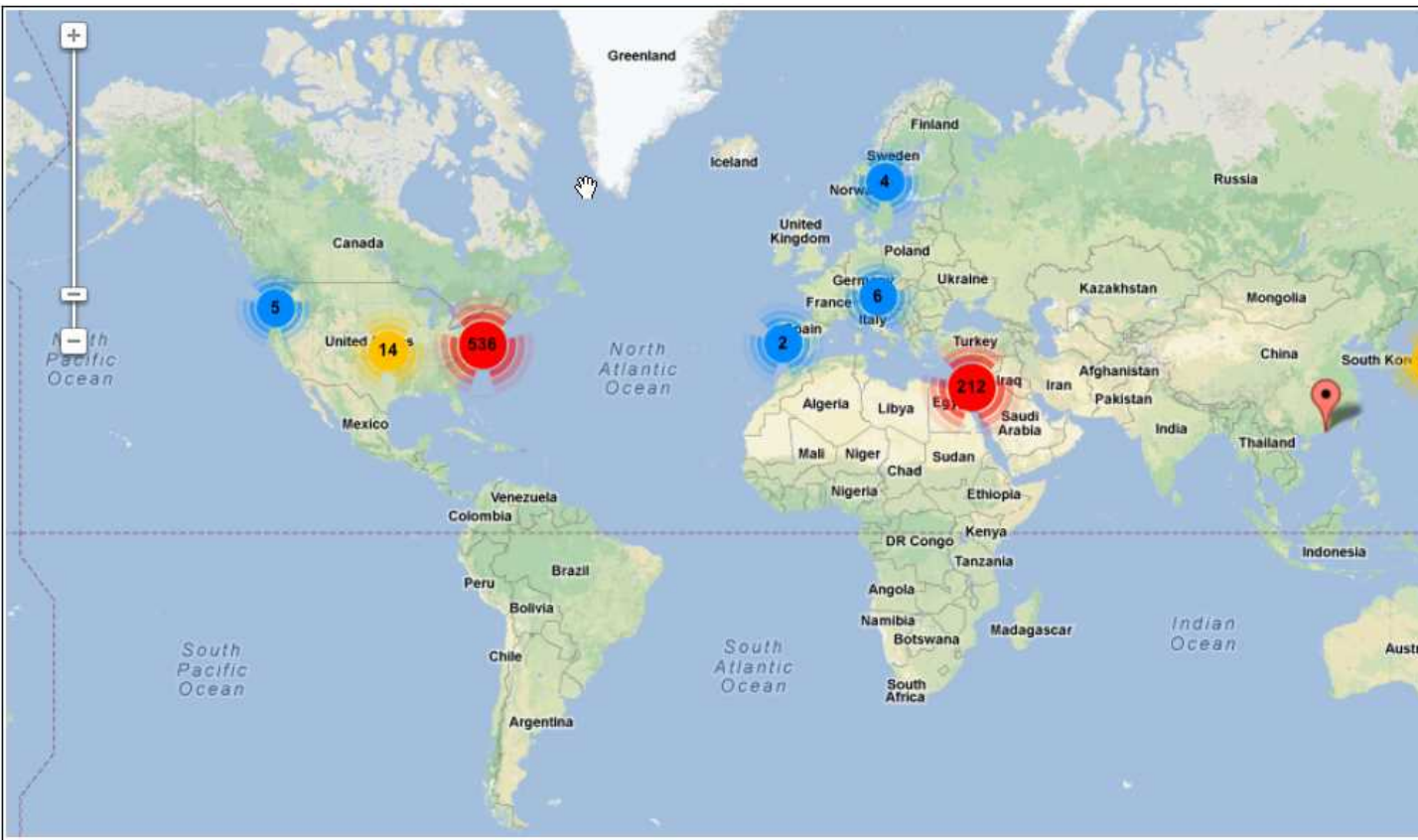
```
2012-12-13 10:29:17,037 -0800 INFO [hostId=prod-frontend-1]  
[module=SERVICE]
```

```
[logger=service.endpoint.auth.v1.impl.AuthenticationServiceDelegate  
[thread=btpool10-8] [remote_ip=67.180.85.25] Successful login for user  
'da@users.com', organization: '000000000000000005
```

Using the example log, running a query like this:

```
| parse "remote_ip=*" as remote_ip  
| lookup latitude, longitude, country_code, country_name, region, city, postal_code, area_code, metro_code  
from geo://default on ip = remote_ip  
| count by latitude, longitude, country_code, country_name, region, city, postal_code, area_code, metro_code  
| sort _count
```

would produce the following results:



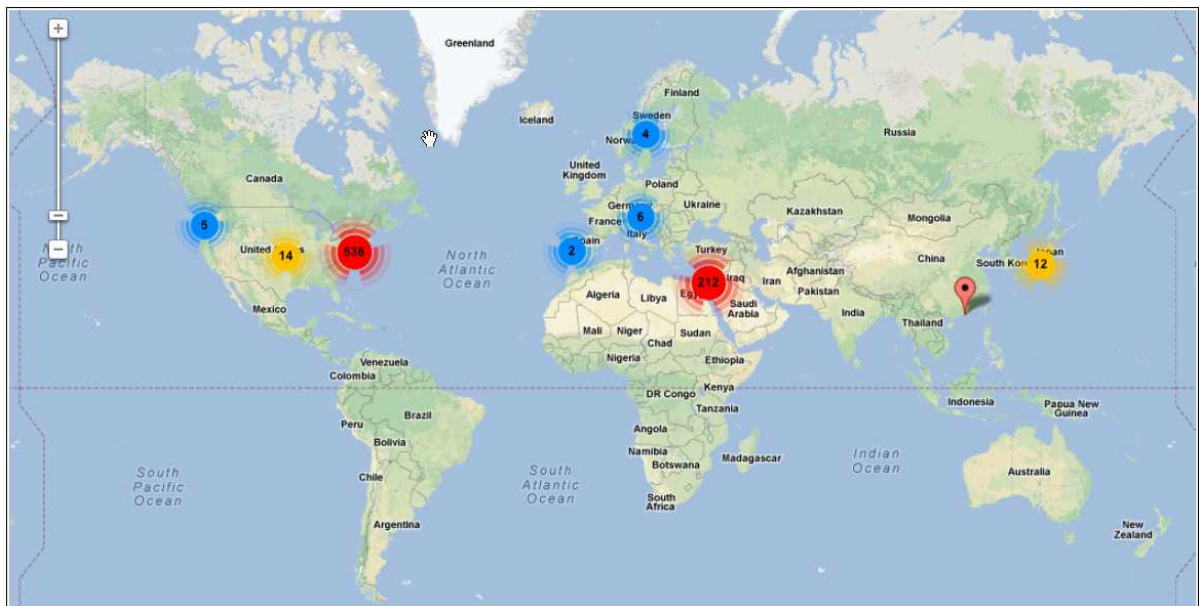
Running a geo lookup query

Enter a query that parses the IP field from your logs, a **lookup** operator to match IP addresses to a lookup table, and then the geo-location fields you'd like to use to chart each IP address.

1. Run a geo lookup query. By default, results display as a table:

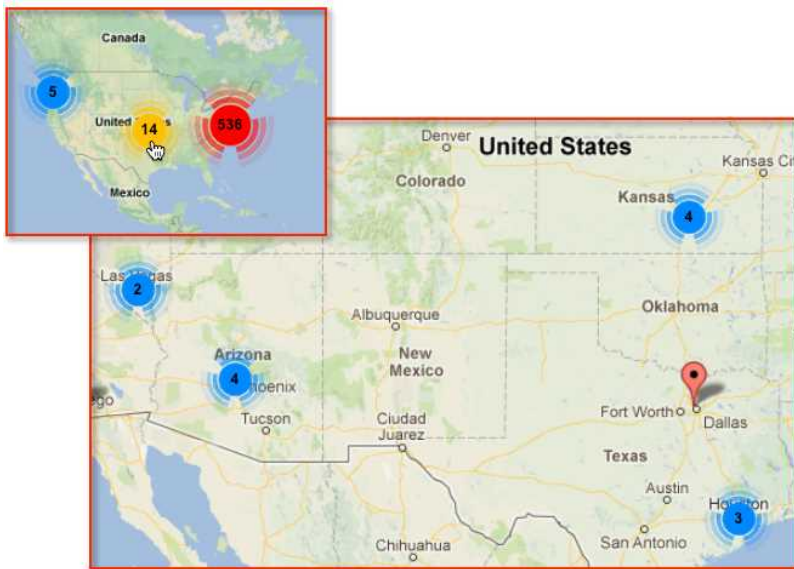
Messages										
Aggregates										
Page: 1 of 1										
#	latitude	longitude	country_code	country_name	region	city	postal_code	area_code	metro_code	_count
1	39.15880	-75.49410	US	United States	DE	Dover	19901	302	504	498
2	31.50000	34.75000	IL	Israel				0	0	212
3	35.64110	-79.84310	US	United States	NC			0	0	35
4	36	138	JP	Japan				0	0	12
5	62	15	SE	Sweden				0	0	4
6	38	-97	US	United States				0	0	4
7	33.67480	-111.95190	US	United States	AZ	Phoenix	85054	480	753	4
8	47.33330	13.33330	AT	Austria				0	0	4
9	29.75230	-95.36700	US	United States	TX	Houston	77002	713	618	2
10	27.99870	-82.51560	US	United States	FL	Tampa	33614	813	539	2
11	36.08771	-115.14850	US	United States	NV	Las Vegas	89119	702	839	2
12	45.51711	-122.68020	US	United States	OR	Portland	97205	503	820	2
13	39.50000	-8	PT	Portugal				0	0	2
14	54	-2	GB	United Kingdom				0	0	2
15	32.80721	-117.16490	US	United States	CA	San Diego	92111	858	825	1
16	22.53329	114.13330	CN	China	30	Shenzhen		0	0	1
17	32.92990	-96.83530	US	United States	TX	Dallas	75244	972	623	1
18	37.78979	-122.39420	US	United States	CA	San Francisco	94105	415	807	1
19	37.42500	-121.94600	US	United States	CA	San Jose	95134	408	807	1
20	40.74210	-74.00180	US	United States	NY	New York	10011	212	501	1
21	29.74190	-95.56430	US	United States	TX	Houston	77042	713	618	1

2. Click the **Map** icon in the **Aggregates** tab. The map displays:

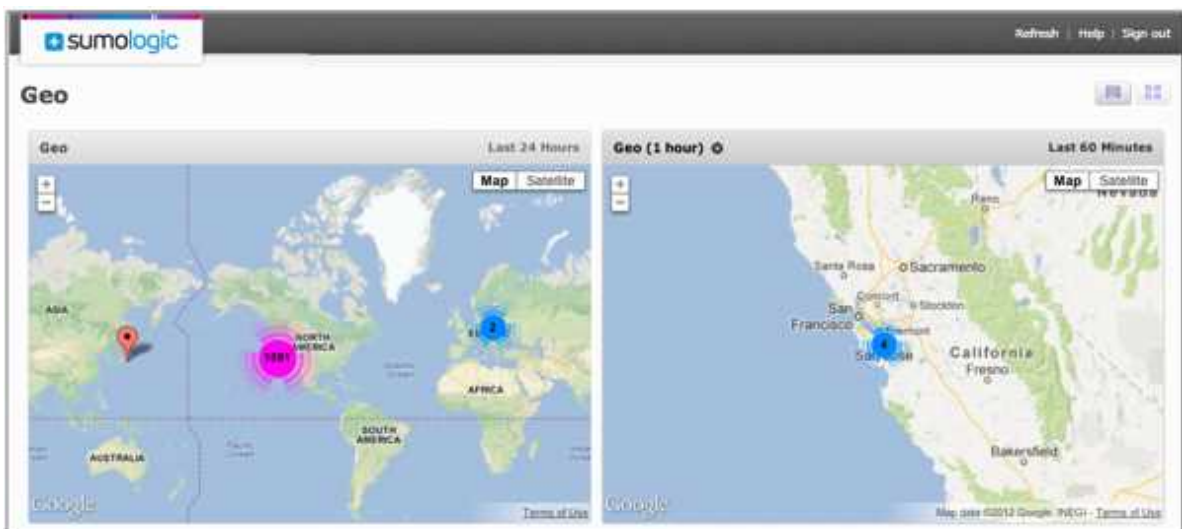


3. Do any of the following:

- Use the zoom slider to zoom in or out on an area of the map. Alternately, click and drag to zoom in or see different areas of a map.
- Click **Satellite** or **Map** to view a satellite or map view.
- Click any marker on the map to see more detail about where IPs originate in a specific area:



4. (Optional) Click **Add to Dashboard** to create a new Dashboard or add the map to an existing Dashboard.



After adding a map to a Dashboard you'll still be able to zoom in and drill down on the data.

Handling null values

To find a mismatch from a geo lookup operator query, use the [isNull operator](#).

For example, running a query like:

```
| parse "remote_ip=*" as remote_ip
| lookup country_code from geo://default on ip = remote_ip
| if (isNull(country_code), "unknown", country_code) as country_code
```

returns results similar to:

Messages				
<div> <div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div>Page: 1 of 1726</div> <div> <div></div> <div></div> </div> <div>LogReduce</div> </div> </div> </div>				
#	Time	remote_ip	country_code	Message
1	05/28/2013 14:05:37.056	*	unknown	2013-05-28 14:05:37,056 running engine: [roleId: definition:Definition(i "remote_ip=*" as remot [{"t":"relative","d":-8 Host: nite-cq-1 ▼ Name: /usr
2	05/28/2013 14:05:37.053	*	unknown	2013-05-28 14:05:37,053 running engine: [roleId: definition:Definition(i latitude, longitude, co sort_count, timerange: Host: nite-cq-1 ▼ Name: /usr
3	05/28/2013 14:05:37.052	*	unknown	2013-05-28 14:05:37,052 running engine: [roleId: definition:Definition(i "remote_ip=*" as remot [{"t":"relative","d":-3 Host: nite-cq-1 ▼ Name: /usr
4	05/28/2013 14:05:37.047	*	unknown	2013-05-28 14:05:37,047 running engine: [roleId: definition:Definition(i "remote_ip=*" as remot [{"t":"relative","d":-6 Host: nite-cq-1 ▼ Name: /usr
5	05/28/2013 14:05:37.039	*	unknown	2013-05-28 14:05:37,039 running engine: [roleId: definition:Definition(i "remote_ip=*" as remot [{"t":"relative","d":-6 Host: nite-cq-1 ▼ Name: /usr
6	05/28/2013 14:05:37.037	*	unknown	2013-05-28 14:05:37,037 running engine: [roleId: definition:Definition(i ! "logger=service.util.i "test@demo.com" parse [{"t":"relative","d":-6 Host: nite-cq-1 ▼ Name: /usr
7	05/28/2013 14:05:37.035	*	unknown	2013-05-28 14:05:37,035 running engine: [roleId: definition:Definition(i country_code, country_n country_code, country_n predicate: , organizati Host: nite-cq-1 ▼ Name: /usr
8	05/28/2013 14:05:37.034	*	unknown	2013-05-28 14:05:37,034 running engine: [roleId: definition:Definition(i latitude, longitude, co sort_count, timerange:

IPv4ToNumber Operator

The `ipv4ToNumber` operator allows you to convert an Internet Protocol version 4 (IPv4) IP address from the octet dot-decimal format to a decimal format. This decimal format makes it easier to compare one IP address to another, rather than relying on IP masking.

Syntax:

- `ipv4ToNumber(ip_addr) as ip_number`

Rules:

- The input to the function must be a valid IPv4 address string.

Example:

Parse IP addresses and convert to number.

The following query parses IP addresses, and converts them to numbers, then uses the `fields` operator to remove all fields except "ip" and "num".

```
_sourceCategory=service remote_ip
| parse "[remote_ip=*)" as ip
| ipv4ToNumber(ip) as num
| fields ip, num
```

would produce results like:

#	Time	ip	num	Message
1	06/20/2014 12:46:59.839	10.163.3.1	178,455,311	2014-06-20 12:46:59.839 [logger=katta] starting connection to Host: nite-search-1

Detect the IP range for a single user.

The following query looks at the number of IP addresses, and the IP range, by user. This could be used to determine if someone has hacked a user account.

```

_sourceCategory=service remote_ip
| parse "auth=User:*" as user
| parse "[remote_ip=*" as remote_ip
| ipv4ToNumber(remote_ip) as remote_ip_dec
| max(remote_ip_dec) as max_ip, min(remote_ip_dec) as min_ip, count_distinct(remote_ip_dec) as
count_ips by user
| max_ip - min_ip as ip_range
| where ip_range > 0
| fields user, count_ips, ip_range

```

would produce results like:

#	user	count_ips	ip_range
1	sanity_test@demo.com	2	2,559,574
2	-----@demo.com	3	3,019,365,458
3	-----@demo.com	2	1
4	l-----r@demo.com	2	1,282,588,031
5	-----:@demo.com	2	1,068,822,004

isNull Operator

The **isNull** operator takes a single parameter and returns a Boolean value: True if the variable is indeed null, or false if the variable contains a value other than null.

Fields can hold a null value for the following reasons:

- There is a mismatch from a lookup operator query.
- There is a missing field from a geo lookup operator query.
- There is a missing field from a transpose operator query.

Syntax:

...isNull(a)...

Examples:

Run a geo lookup query where we can find remote IP addresses that are not in the geo database. In this situation, no country_code will be associated with the IP address (this field would be null).

Running a query like:

```

| parse "remote_ip=*" as remote_ip
| lookup country_code from geo://default on ip = remote_ip
| if (isNull(country_code), "unknown", country_code) as country_code

```

returns results similar to:

Join Operator

The **Join** operator combines records of two or more data streams. Results are admitted on-the-fly to allow real time tables to be built. Values common to each table are then delivered as search results.

The Join operator in Sumo Logic works much like a standard SQL join.

Syntax:

```
[search terms] | join
(parse "starting stream from *" AS a) as t1,
(parse "starting search * from parent stream *" AS b, c) as t2,
(parse "starting save * from parent stream *" AS d, e) as t3
on t1.a = t2.c
and t1.a = t3.e
```

A **timewindow** can be added to constrain how far apart in time records are allowed to join. The syntax for adding a timewindow is:

```
[search terms] | join
(parse "starting stream from *" AS a) as t1,
(parse "starting search * from parent stream *" AS b, c) as t2,
(parse "starting save * from parent stream *" AS d, e) as t3
on t1.a = t2.c
and t1.a = t3.e
[timewindow 10m]
```

To operate on fields in each table after the ON clause, use this syntax:

```
[search terms] | join
(parse "starting stream from *" AS a) AS t1,
(parse "starting search * from parent stream *" AS b, c) AS t2
on t1.a = t2.c
| fields t1_a, t2_b
```

Rules:

- Two or more tables must be created for a query.
- Data must be present in the time range you choose for the query.
- Queries that include a Join operator cannot be added to Dashboards.
- Only conjunctive conditions (AND) are allowed. Using NOT or OR conditions is not supported.
- The following conditions are not currently supported in the ON clause:
 - t1.a = 3
 - t1.a != t2.c
 - NOT t1.a
 - t1.a = t2.c OR t1.b = t2.d

- Sub queries are supported, and can include aggregate operators.

Examples

Running a Join operator query.

For this example we'll run a Join query on two tables using logs that look like:

```
starting stream from stream-2454
starting stream from stream-7343
starting search search-733434 from parent stream stream-2454
starting search search-854343 from parent stream stream-7343
starting stream from stream-6543
starting search search-455563 from parent stream stream-6543
starting search search-32342 from parent stream stream-7343
```

Running a query like:

```
* | join
(parse "starting stream from *" AS a) AS T1,
(parse "starting search * from parent stream *" AS b, c) AS T2
on T1.a = T2.c
```

returns results similar to:

a	b	c
stream-2454	search-733434	stream-2454
stream-7343	search-854343	stream-7343
stream-7343	search-32342	stream-7343
stream-6543	search-854343	stream-6543

Using Join with a Diff operator.

Let's say our logs look something like:

```
event=login session=12345 time=20130512
event=purchase session=12345 value=50
event=login session=23456 time=20130513
event=purchase session=12345 value=100
event=purchase session=23456 value=120
event=purchase session=23456 value=200
event=purchase session=23456 value=20
```

Running a query like:

```
* | join
(parse "event=login session=* time=*" AS s1,time) as t1,
```

```
(parse "event=purchase session=* value=*" AS s2, v2) as t2
on t1.s1 = t2.s2
```

Produces results similar to:

s1	time	s2	v2
12345	20130512	12345	50
12345	20130512	12345	100
23456	20130513	23456	120
23456	20130513	23456	200
23456	20130513	23456	20

Adding a Diff operator, such as:

```
* | join
(parse "event=login session=* time=*" AS s1,time) as t1,
(parse "event=purchase session=* value=*" AS s2, v2) as t2
on t1.s1 = t2.s2
join terms | diff v2 by s2
```

produces results similar to:

s1	time	s2	v2	_diff
12345	20130512	12345	50	null
12345	20130512	12345	100	50
23456	20130513	23456	120	null
23456	20130513	23456	200	80
23456	20130513	23456	20	-180

Operate on fields after the ON clause.

Assume you have a Join query such as this:

```
* | join
(parse "starting stream from *" AS a) AS t1,
(parse "starting search * from parent stream *" AS b, c) AS t2
on t1.a = t2.c
```

After the Join statement, to use the T1.a and the T2.b fields in subsequent clauses, you would instead refer to them as T1_a and T1_b. For example, to use the Fields operator to single out the T1.a and T2.b values, use the following query:

```
* | join
(parse "starting stream from *" AS a) AS t1,
(parse "starting search * from parent stream *" AS b, c) AS t2
on t1.a = t2.c
fields t1_a, t2_b
```

JSON Operator

The **JSON** operator is a search query language operator that allows you to extract values from JSON input. Because JSON supports both nested keys and arrays that contain ordered sequences of values, the Sumo Logic JSON operator allows you to extract:

- Single, top-level fields.
- Multiple fields.
- Nested keys.
- Keys in arrays.

The JSON operator also supports the **nodrop** option, which allows messages containing invalid JSON values to be displayed.

The following examples use this sample log message:

```
2014-03-11 15:00:42,611 -0700 INFO [hostId=prod-search-6]
[explainJsonPlan.stream] {"module":"stream","logMessage":"exiting
search","sessionId":"90D97000","customerId": "00B12CD0"
...
{
  "baselineIntervals":[
    "2014-03-11T23:00:00.000-07:00\2014-03-12T05:00:00.000-07:00",
    "2014-03-12T05:00:00.000-07:00\2014-03-12T11:00:00.000-07:00"],
  "meta":{
    "type": "timestamps",
    "version": "1"
  }
}
...
```

Extracting a single top-level field

The JSON operator allows you to extract a single, top-level field. For example, to extract `sessionId`:

```
_sourceCategory=stream RawOutputProcessor "\"message\""
| parse "explainJsonPlan.stream]" as jsonobject
| json field=jsonobject "sessionId"
| fields -jsonobject
```

produces results like:

#	Time	sessionId	Message
1	03/11/2014 14:21:23.085	A199B96B04993284	

Extracting multiple fields

You can also extract multiple fields in a single operation. For example, to extract `sessionId` and `customerId`:

```
_sourceCategory=stream RawOutputProcessor "\"message\""  
| parse "explainJsonPlan.stream[*]" as jsonobject  
| json field=jsonobject "sessionId", "customerId"  
| fields -jsonobject
```

produces these results:

#	Time	sessionId	customerId	Message
1	03/11/2014 14:18:36.176	B517E16BBCEB9FEB	00000000001ECE6B	2014-03-11 14:18:36,176 -0700 [REDACTED] module : stream , logMessage

In addition, you can assign names to fields that differ from their assigned names. To use `sd` instead of `sessionId` and `cid` instead of `customerId`, like this:

```
_sourceCategory=stream RawOutputProcessor "\"message\""  
| parse "explainJsonPlan.stream[*]" as jsonobject  
| json field=jsonobject "sessionId", "customerId" as sID, cID  
| fields -jsonobject
```

which gives you these results:

#	Time	sd	cid	Message
1	03/11/2014 14:32:11.200	C13005303267A1DF	00000000004EF771	2014-03-11 14: [REDACTED]

Extracting a nested key

The example log message has nested keys, which you can extract by specifying the path using dot notation:

For example, to extract the nested key `type` from `meta` , use the following query:

```
* | json field=jsonobject "meta.type"
```

Finding values in a JSON array

In some cases, fields values are actually arrays, like `baselineIntervals` in the example log message:

You can instruct the JSON operator to extract `@baselineIntervals`, like this:

```
* | json field=jsonobject "baselineIntervals"
```

It returns a list of the values in the array: `["2014-03-10T23:...", "2014-03-11T05:..."]`.

like this:

#	Time	jsonobject	baselineintervals
1	03/11/2014 16:07:14.247	{"baselineIntervals": ["2014-03- 10T23:00:00.000- 07:00/2014-03- 11T05:00:00.000- 07:00","2014-03- 11T05:00:00.000- 07:00/2014-03- 11T11:00:00.000- 07:00"]}	["2014-03- 10T23:00:00.000- 07:00/2014-03- 11T05:00:00.000- 07:00","2014-03- 11T05:00:00.000- 07:00/2014-03- 11T11:00:00.000- 07:00"]



To refer to one specific entry in the array, provide the array's index: * | json field=jsonobject
"baselineIntervals[1]"

Using the nodrop option

By default, the JSON operator optimizes results by dropping messages that don't use the specified key or keys, or messages that use invalid JSON keys. Use the **nodrop** option to prevent this optimization, and set the extracted field values to null (empty):

```
* | json field=jsonobject "baselineIntervals[0]" nodrop
```

Keyvalue operator

Typically, log files contain information that follow a key-value pair structure. The **keyvalue** operator allows you to get values from a log message by specifying the key paired with each value.

For example, a log could contain some the following keys (highlighted):

```
2012-06-17 17:02:08,880 -0700 INFO [hostId=prod-frontend-5] [module=SERVICE]
[logger=service.endpoint.search.v1.impl.SearchServiceImpl] [thread=Thread-797 (group:Ho
[auth=User:test@demo.com:000000000000BE79:0000000000000005:false] [remote_ip=50.18.185.
[session=B046486365A098F9] [customer=0000000000000005] [call=OutboundStreamProtocol.rec
Count update for stream session: 'B046486365A098F9', stream query: 'B046486365A098F9', a
```

From that log message, you can use the **keyvalue** operator to get the values for one or more keys. For example, if you'd like to see information just about the "remote_ip" value, running this query:

```
... | keyvalue "remote_ip" ...
```

would produce these results:

#	Time	remote_ip	Message
1	06/27/2012 17:42:25.732	50.18.75.232	2012-06-27 17:42:25,732 -0700 INFO [hostId=nite-frontend-1] [module=RECEIVER] [logger=scala.receiver.MessageB [thread=MTP-MessagePilePipeline-14] [auth=Collector:nite-hornetq-inbound-2:00000000000A5F71:000000000000168:f [web_session=BxmQBHC2...] Pile for customer: '000000000000168', ID: '800000004D06553', block: '8000000000108 collector: '00000000000A5F71' Host: nite-frontend-1 Name: /usr/sumo/receiver-19.0-994/logs/receiver.log Category: receiver
2	06/27/2012 17:42:25.623	184.72.8.220	2012-06-27 17:42:25,623 -0700 INFO [hostId=nite-frontend-1] [module=RECEIVER] [logger=scala.receiver.MessageB [thread=MTP-MessagePilePipeline-16] [auth=Collector:nite-hornetq-mgmt-1:000000000000259:000000000000168:fals [web_session=MFVxOFYA...] Pile for customer: '000000000000168', ID: '800000004D06552', block: '8000000000108 collector: '000000000000259' Host: nite-frontend-1 Name: /usr/sumo/receiver-19.0-994/logs/receiver.log Category: receiver
3	06/27/2012 17:42:25.468	204.236.183.62	2012-06-27 17:42:25,468 -0700 INFO [hostId=nite-frontend-1] [module=RECEIVER] [logger=scala.receiver.MessageB [thread=MTP-MessagePilePipeline-15] [auth=Collector:nite-frontend-1:00000000000023D:000000000000168:false] [br/>[web_session=EWFoPaCM...] Pile for customer: '000000000000168', ID: '800000004D06551', block: '8000000000108 collector: '00000000000023D' Host: nite-frontend-1 Name: /usr/sumo/receiver-19.0-994/logs/receiver.log Category: receiver

The keyvalue operator can also be used in two explicit modes: the default inference mode, and Regular Expression mode.

Inference mode syntax

When used in the default **inference mode**, the **keyvalue** operator uses an internal list of regular expressions to determine how to extract the value for a given key. This greatly simplifies the syntax.

For example, you could extract the keys "module" and "thread" and their values from a log message by running this query:

```
* | keyvalue infer "module", "thread"
```

to produce these results:

#	Time	module	thread	Message
1	03/19/2014 13:26:58.825	RECEIVER	1686299103@qtp-93187164-7168	2014-03-19 13:26:58.825 [logger=scala.receiver.MessageB 12:0000000005F51 [web_session=i9: size = 1030

Regular Expression mode syntax

In **Regular Expression mode**, you must explicitly match keys and values based on a regular expression. This allows for greater flexibility than inference mode. For example, to extract the values for the keys "serviceinfo_IP", "loggingcontext.region", and "request.method" fields from a log message, use this query:

```
* | keyvalue regex "=(.*?)[,;]" keys "serviceinfo.IP", "loggingcontext.region", "request.method" as ip, region, method
```

This provides the following results:

#	Time	ip	region	method	Message
1	03/19/2014 13:40:25.010	'Finished successfully'	'Finished successfully'	'Finished successfully'	2014-03-19 [logger=str [auth=User :

The **keyvalue** operator also supports regular expressions that contain a **single match group**. (You may notice an improvement in performance by running queries with a single match group.) For example, you could run this query to get the same results as the previous query:

```
* | kv regex "=(.*)", "serviceinfo.IP", "loggingcontext.region", "request.method"
```

In the above case, the operator first finds the key itself in the message (first occurrence), and then finds the closest match of the regular expression to the location in the message where the key was found.



The number of fields specified with the "as" clause must match the number of key-value pairs specified. You can omit the clause if you'd like the operator to automatically create the field names for the extracted values. To do this, **keyvalue** replaces every character (other than a..zm A..Z, 0..9, or _) with an underscore (_).

Abbreviated syntax

The **keyvalue** operator can be abbreviated in either mode (Inference or Regular Expression). For example, running this query:

```
* | keyvalue infer keys "a", "b"
```

will produce the same results as running this query:

```
* | keyvalue "a", "b"
```

Also, **keyvalue** can be abbreviated to "kv". For example:

```
* | kv "a", "b"
```

Limit Operator

The limit operator reduces the number of raw messages or aggregate results returned. If you simply query for a particular term, for example "error" without using an aggregation operator such as group by, limit will reduce the number of raw messages returned. If you first use group-by or other aggregation operator, the limit operator will reduce the number of grouped results instead.

The limit operator is useful for creating lists of events for a dashboard, which allows you to see at a glance, for example, the "Top 10" service operations, system operations, errors, or other system or user activities.



Sumo Logic Apps often use the limit operator in queries to display system data in dashboards for various uses.

Syntax:

- ... | limit #

Examples:

Top 10 errors.

In this example, we simply query for the term "error" without using an aggregation operator, and limit will reduce the number of raw messages returned to 10.

error * | limit 10

The message tab displays only the first 10 error messages for the time range you have queried.

Count Top 5 errors for a source.

In this query, you can search for errors, count by the `_sourceCategory`, sort by the count, and limit the results to the top 5 errors.

error * | count by _sourceCategory | sort by _count | limit 5

which would provide results similar to:

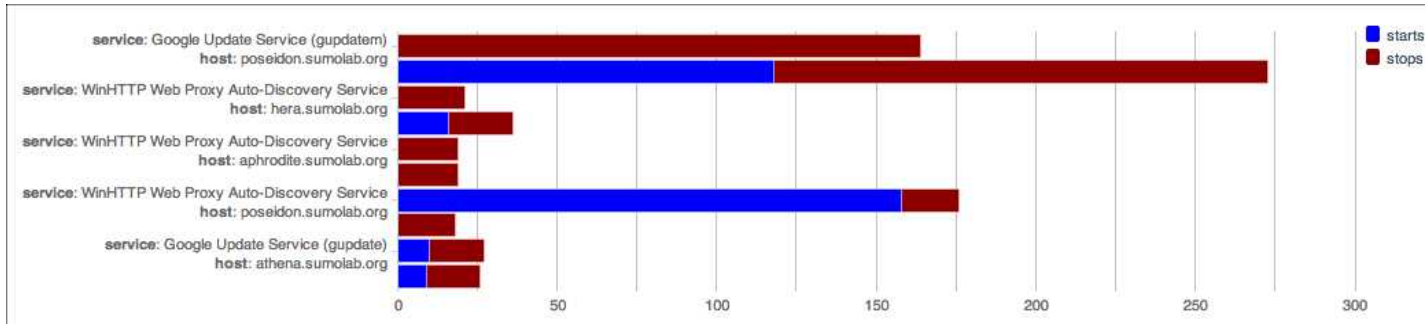
#	_sourcecategory	_count
1	esx_perf	2,080
2	CommVault	909
3	OS/Windows	779
4	OS/Linux/Security	337
5	Symantec/AntiVirus	122

Top 10 Service Operations:

In this query, you can see the top 10 Windows services per host that have started and stopped over the last 10 hours.

```
_sourceCategory=OS/Windows Service Control Manager
| parse regex "Message = \"The (?<service>\w.+?) service entered the (?<state>\w+) state\"
| parse regex "ComputerName = \"(?<host>[^\"]+)\";"
| if(state=="running", 1, 0) as starts
| if(state=="stopped",1,0) as stops
| sum(starts) as starts, sum(stops) as stops by service,host
| sort by stops, starts
| limit 10
```

which can be displayed in a bar chart like this:



See [Sort Operator](#) for more information.

Lookup Operator

Using a lookup operator, you can map data in your log messages to meaningful information. For example, you could use a lookup operator to map "userID" to a real user's name. Or, you could use a lookup operator to find black-listed IP addresses. In either case, you'll point the operator to one of the following:

- A table of saved data generated by the [Save operator](#).
- A CSV file that describes the relationship between the log file message and an external source.

Syntax

A lookup operator follows this form:

lookup [outputColumns] from [filePath] on [joinColumns]

where:

- outputColumns is a list of field names in the header of the filePath.
- filePath is an HTTPS address of a CSV file containing the external relationship table or a table saved to the Sumo Logic file system by the Save operator.

Note: HTTP basic authentication is supported for .csv files, with the following syntax:

https://USERNAME:PASSWORD@company.com/userTable.csv

- joinColumns is a list that defines the relationship between values in the log data with matching values in an external table.



For limitations, see [Using Lookup to Access Saved Data](#), in Beta Features.

Structuring CSV Files

For details on CSV file requirements, see [Structuring CSV Files](#).

Examples

Type the lookup operator in the **Search** tab of the Sumo Logic Web Application, just as you would any other operator.

To match the userID string with a users' ID in your CSV, your query could be:

```
* | parse "name=*, phone number=*" as (name, phone) | lookup email from
https://compay.com/userTable.csv on name=userName, phone=cell
```

where the `userTable.csv` file includes the following:

```
"id", "userName", "email", "IP", "cell"
"1", "Joe", "joe@example.com", "192.168.1.1", "650-123-4567"
"2", "John", "john@example.com", "192.168.1.2", "212-123-4567"
"3", "Susan", "susan@example.com", "192.168.1.3", "914-123-4567"
"4", "John", "another_john@example.com", "192.168.1.4", "408-123-4567"
"5", "John", "yet_another_john@example.com", "192.169.1.5", "734-123-4567"
```

Running this query adds three fields to the output: **userName**, **email**, and **IPAddress**.

Composite field lookup

In our example above we had several users named John. A lookup operator can be used on a composite set of fields, so you can identify the correct email for each person named John because each unique cell phone number has also been mapped using a query like:

```
* | parse "name=*, phone number=*" as (name, phone)
| lookup email from https://compay.com/userTable.csv on name=userName, phone=cell
```

Running this query adds an **email** field to the output.

Using multiple lookup operators together

Another way to use a lookup operator is to chain lookup operators together. Each operator can call separate `.csv` files. For example, if you wanted to find user names and the position each user has in a company, your query could be:

```
* | parse "userID=*" as userID | lookup userName from https://company.com/userTable.csv on userID=id
| lookup position from https://company.com/userPosition.csv on userID=id
```

where the `userPosition.csv` file includes the following:

```
"id", "position"
"1", "Salesperso"
"2", "Salesperson"
"3", "Engineer"
"4", "Manager"
```

```
"5", "Senior Engineer"
```

In our example above, the first operator finds the name, and the second finds the position.

Handling null values

To find a mismatch from a lookup operator query, use the isNull operator.

For example, running a query like:

```
| parse "remote_ip=*" as remote_ip  
| lookup country_code from geo://default on ip = remote_ip  
| if (isNull(country_code), "unknown", country_code) as country_code
```

returns results similar to:

characters that are not numerals, and checks if the resulting string is a valid credit card number, returning true or false accordingly.

Syntax:

- ...| luhn(input: String) as isValid (boolean)
- ...| luhn("0000000000000000") as ccnumber

Examples:

Identify and verify credit card numbers in message logs.

Use the following query to identify credit card numbers in message logs, and verify them using the Luhn operator:

```
| parse regex "(?<maybecc>\d{4}-\d{4}-\d{4}-\d{4})" nodrop
| parse regex "(?<maybecc>\d{4}\s\d{4}\s\d{4}\s\d{4})" nodrop
| parse regex "(?<maybecc>\d{16})" nodrop
| if (luhn(maybecc), true, false) as valid
```

which provides results such as:

#	Time	hostid	maybecc	valid	Message
204,121	05/14/2014 13:11:26.034	nite-mix-1	00000000000000131	false	2014-05-14 13:11:26.034 [localUserNan [thread=Threa [auth=Custom SystemUser:3] received star 00000000000000 Host: nite-mix-1

Search for and verify a specific credit card number.

Use the following query to search for a specific credit card number and verify it using the Luhn operator:

```
*| "6666-7777-6666-8888" as b | luhn(b) as d
```

It would provide the following results:

#	Time	b	d	Message
1	05/13/2014 13:10:02.000	6666-7777-6666-8888	true	2014-05-13 Host: 54.187.1
2	05/13/2014 13:10:02.000	6666-7777-6666-8888	true	2014-05-13 Host: 54.187.1

Matches Operator

The **matches** operator can be used to match a string to a pattern. The return of the operator is boolean; the operator can be used with **where** or **if** expressions.

Matches operators can be used in Dashboard Monitors, and are very commonly used in conjunction with other operators to build robust queries.

Syntax:

- ... [string expression] matches [pattern] ...
- ... if [string expression] matches [pattern] ...
- ... where [string expression] matches [pattern]...
- ... ! [string expression] matches [pattern]

Examples

Identifying the browsers and operating systems used to access your website

Running a query containing a matches operator on Apache Access logs can show you the breakdown of the devices and browsers that are accessing your site. You can then create a Dashboard with this query. We've used a transpose operator in this query to allow us to name the axis of our column chart.

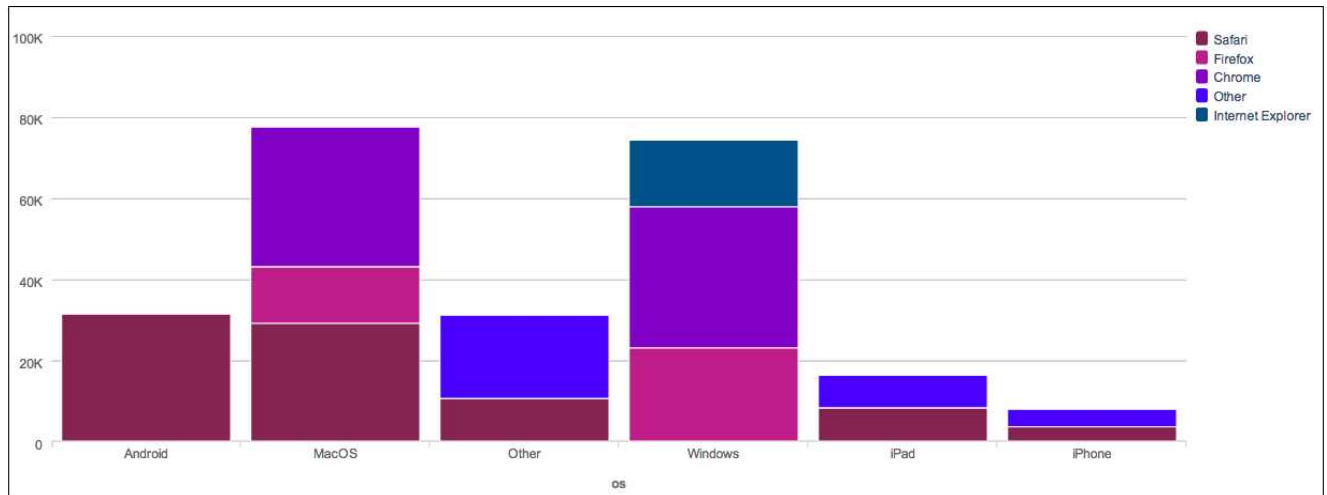
Running a search like:

```
_sourceCategory=Apache/Access
| extract "\"[A-Z]+ \S+ HTTP/[d.]+\" \S+ \S+ \S+ \"(?<agent>[^\"]+?)\""
| if (agent matches "**Windows NT**","Windows","Other") as OS
| if (agent matches "**Macintosh**","MacOS",OS) as OS
| if (agent matches "**iPad**","iPad",OS) as OS
| if (agent matches "**iPhone**","iPhone",OS) as OS
| if (agent matches "**Android**","Android",OS) as OS

| if (agent matches "**MSIE**","Internet Explorer","Other") as Browser
| if (agent matches "**Firefox**","Firefox",Browser) as Browser
| if (agent matches "**Safari**","Safari",Browser) as Browser
| if (agent matches "**Chrome**","Chrome",Browser) as Browser

| count(agent) by OS,Browser
| transpose row os column browser as *
```

Produces aggregate results similar to:



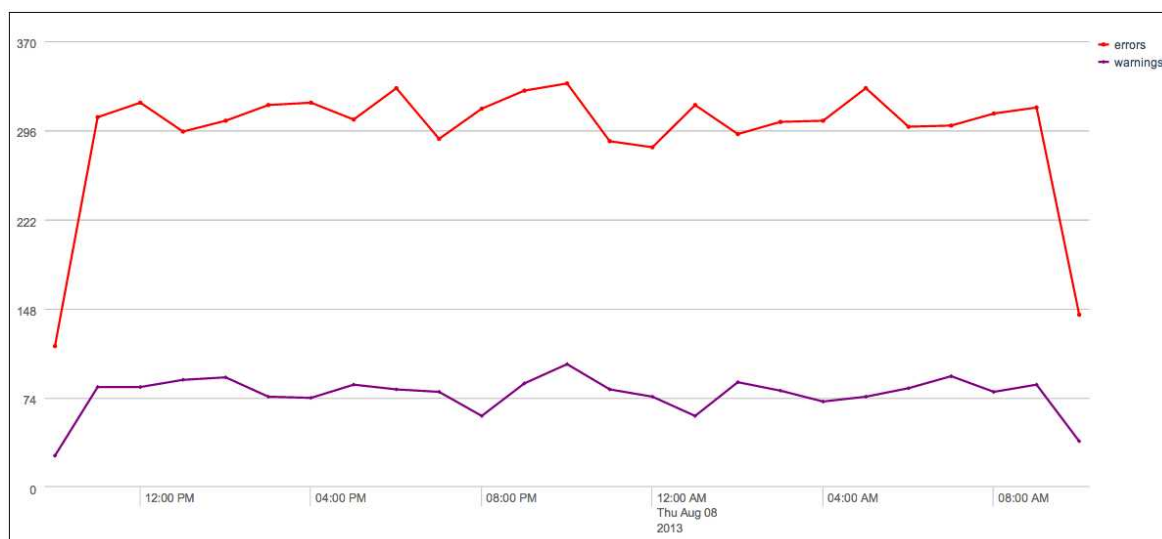
Viewing errors and warnings over time

In this example, we'll run a query against Windows logs to see the distribution of errors and warnings over the previous hours. Using a timeslice operator in the query breaks the results into one hour buckets.

Running a search like:

```
_sourceCategory=OS/Windows (error or warning)  
| parse "Type = \"\";" as evtType  
| if (_raw matches "**EventType = Error*",1,0) as errors  
| if (_raw matches "**EventType = Warning*",1,0) as warnings  
| if (evtType matches "Error*",1,errors) as errors  
| if (evtType matches "Warning*",1,warnings) as warnings  
| timeslice by 1h  
| sum(errors) as errors, sum(warnings) as warnings by _timeslice  
| sort _timeslice asc
```

Produces results similar to:



For more examples, see [Refining a Query](#).

Math Expressions

You can use general mathematical expressions on numerical data extracted from log lines. For any mathematical or group-by function that implicitly requires integers, Sumo Logic casts the string data to a number for you.

Syntax:

- ... | expression [as alias] | ...

Rules:

- The term "expression" is evaluated as a mathematical expression in the context of existing fields.
- Parentheses can be used to group operations.
- The ternary operator is supported so you can use "condition ? value_if_true : value_if_false".
- Supported mathematical operators +, -, *, /, %

Examples:

- Boolean expression tests like: `disk > 0.8 ? 1 : 0` as overcapacity
- Math function calls like: `min((fps / 10 + 1) * 10, 70)` as bucket
- Assuming $x = 1, 2$, then
`ceil(sqrt(x*x + y*y))` as d
should result in $d = 2.0$

Basic Functions

Standard math functions are supported.

abs

Calculates the absolute value of x.

Syntax:

- `abs(x)`

Example:

- `abs(-1.5) as v // v = 1.5`

round

Returns the closest integer to x.

Syntax:

- `round(x)`

Example:

- `round((bytes/1024)/1024) as MB`

ceil

Rounds up to the smallest integer value. Returns the smallest integral value that is not less than x.

Syntax:

- `ceil(x)`

Examples:

- `ceil(1.5) as v // v = 2`
- `ceil(-1.5) as v // v = -1`

floor

Rounds down to the largest previous integer value. Returns the largest integer not greater than x.

Syntax:

- `floor(x)`

Examples:

- `floor(1.5) as v // v = 1`
- `floor(-1.5) as v // v = -2`

max

Returns the larger of two values.

Syntax:

- `max(x, y, z)`

Example:

- `max(1, 2) as v // v = 2`

min

Returns the smaller of two values.

Syntax:

- `min(x, y, z)`

Example:

- `min(1, 2) as v // v = 1`

sqrt

Returns the square root value of x.

Syntax:

- `sqrt(x)`

Example:

- `sqrt(4) as v // v = 2`

cbrt

Returns the cube root value of x.

Syntax:

- `cbrt(x)`

Example:

- `cbrt(8) as v // v = 2`

Exponents and Logs

Exponential and logarithmic functions are supported.

exp

Returns Euler's number e raised to the power of x.

Syntax:

- `exp(x)`

Example:

- `exp(1) as v // v = 2.7182818284590455`

expm1

Returns value of x in $\exp(x)-1$, compensating for the roundoff in $\exp(x)$.

Syntax:

- `expm1(x)`

Example:

- `expm1(0.1)` as `v // v = 0.10517091807564763`

log

Returns the natural logarithm of x.

Syntax:

- `log(x)`

Example:

- `log(2)` as `v // v = 0.6931471805599453`

log10

Returns the base 10 logarithm of x.

Syntax:

- `log10(x)`

Example:

- `log10(2)` as `v // v = 0.3010299956639812`

log1p

Computes $\log(1+x)$ accurately for small values of x.

Syntax:

- `log1p(x)`

Example:

- `log1p(0.1)` as `v // v = 0.09531017980432487`

Trigonomic Functions

Trigonometric functions are supported.

sin

Sine of argument in radians.

Syntax:

- `sin(x)`

Example:

- `sin(1)` as `v // v = 0.8414709848078965`

cos

Cosine of argument in radians.

Syntax:

- `cos(x)`

Example:

- `cos(1)` as `v // v = 0.5403023058681398`

tan

Tangent of argument in radians.

Syntax:

- `tan(x)`

Example:

- `an(1)` as `v // v = 1.5574077246549023`

asin

Inverse sine; result is in radians.

Syntax:

- `asin(x)`

Example:

- `asin(1)` as `v // v = 1.5707963267948966`

acos

Inverse cosine; result is in radians.

Syntax:

- `acos(x)`

atan

Inverse tangent; result is in radians.

Syntax:

- `atan(x)`

atan2

Four-quadrant inverse tangent.

Syntax:

- `atan2(b, c)`

Example:

- `atan2(0, -1)` as `v // v = pi`

sinh

Hyperbolic sine of argument in radians.

Syntax:

- `sinh(x)`

cosh

Hyperbolic cosine of argument in radians.

Syntax:

- `cosh(x)`

tanh

Hyperbolic tangent of argument in radians.

Syntax:

- `tanh(x)`

A Few More Functions

Additional math functions `hypot` and `signum` are also supported, as well as functions to convert between degrees and radians.

hypot

Returns the square root of the sum of an array of squares.

Syntax:

- `hypot(x, y, z)`

Example:

- `hypot(1, 0)` as `v // v = 1`

signum

Returns an array `y` the same size as `x`, where each element of `y` is:

- 1 if the corresponding element of `x` is greater than zero
- 0 if the corresponding element of `x` equals zero

- -1 if the corresponding element of x is less than zero

toDegrees

Converts angles from radians to degrees.

Syntax:

- toDegrees(x)

Example:

- toDegrees(asin(1)) as v // v = 90

toRadians

Converts angles from degrees to radians.

Syntax:

- toRadians(y)

Example:

- toRadians(180) as v // v = pi

Rollingstd Operator

The **rollingstd** operator provides the rolling standard deviation of a field over a defined window. Rollingstd displays this value in a new column named **_rollingstd**.

The rollingstd operator finds the rolling standard deviation of a field, allowing you to identify changes over time. For example, you could use rollingstd in a query to identify spikes in activity for a Collector, or for a URL in your site. You can use a rollingstd to find compute the average number from the past, to identify changes (larger or smaller) over time.

Two or more data points are needed to get accurate results from a rollingstd operator. If you attempt to find the rollingstd of a single data point the results will automatically be zero.

Syntax:

- .. rollingstd field [, window_length]

Rules:

- An alias for rollingstd is optional. When an alias is not provided, **_rollingstd** is the default alias.
- Specified fields must contain numeric values.
- To add a query that includes a rollingstd operator to a Dashboard, you must add a group by function **before** the rollingstd operator.

Example:

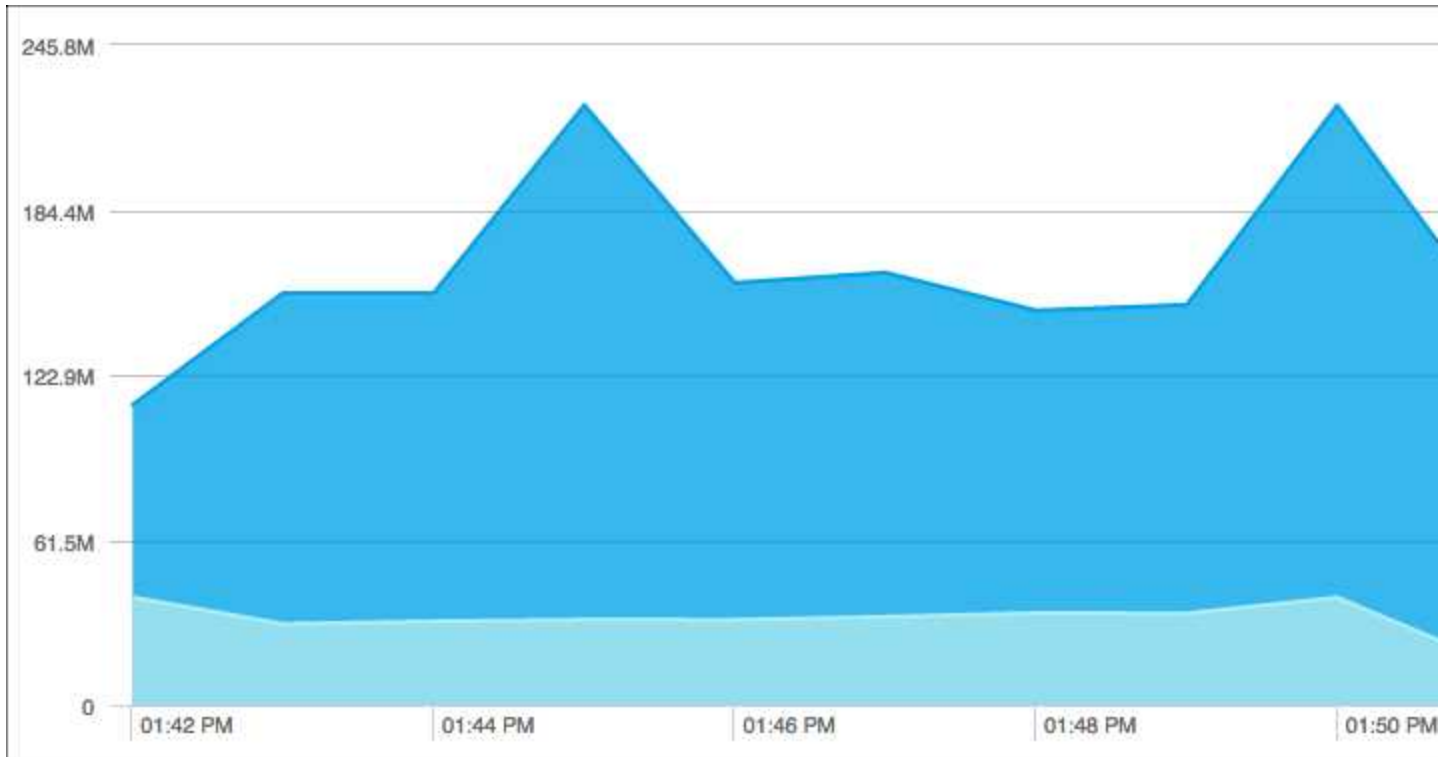
Find the rolling standard deviation of a field between time points. Using rollingstd with timeslice, you can run a query similar to:

`* | parse "bytes: *" as bytes | timeslice 1m | sum(bytes) as bytes by _timeslice | sort _timeslice | rollingstd bytes, 5`

that produces results like:

#	Time	bytes	_rollingstd
1	04-09-2014 13:56:00	20,617,126	0
2	04-09-2014 13:55:00	160,589,841	98,975,655.95759
3	04-09-2014 13:54:00	115,973,570	71,502,708.56728
4	04-09-2014 13:53:00	148,131,832	63,327,969.10010
5	04-09-2014 13:52:00	144,419,816	56,805,247.89935

The aggregation table can be made into an area chart, like this:



Rollingstd is also used with the [Backshift Operator](#).

Save Operator

Using a Save operator allows you to save the results of a query into the Sumo Logic file system. Later, you can use the lookup operator to access the saved data. Save operator saves data in a simple format to a location you choose.



You'll need to remember the path where you point the save operator to put the file. You may want to save searches that contain save operators so you can refer to it later. (At this time there's no way locate the saved file if you forget the path.)

Let's say you want to save data about new user accounts created each day. Your save operator could look like:

```
| parse "name=*" as name
| parse "action=*" as action
| parse "date=*" as date
| where action="sign-up"
| first(date) as date, first(action) as action by name
| save myFolder/mySubFolder/newDailyUsers
```

The above search would create a file that looks a bit like this:

Name	Action	Date
John	sign-up	2012-08-20
Bill	sign-up	2012-08-21
Bob	sign-up	2012-08-21

You can access data in the saved table using the lookup operator.



Aggregate results can also be saved with the save operator.

Saving files to a shared location

A file generated by a save operator can be saved to an org-level shared folder. This allows for others in your organization to use your search results when running their lookup queries.

Note that files saved to a shared location can only be modified by the person who originally shared the file.

To save a file to a shared location:

- Include the following at the end of your query:
...save /shared/myFolder/mySubFolder/fileName

In the path, the word "shared" can be any combination of cases.



For limitations, see [Using Lookup to Access Saved Data](#), in Beta Features.

Appending to saved files

Once you've created a file generated by a save operator, you can append data at any time. If you're running a scheduled daily search that calculates properties for the current day, that data is appended to the existing file containing results from the previous days. Data you append to a file must match exactly; if the new results don't match the previous results an error message appears, including cases where you attempt to append with additional fields.



If you don't use "append" the previously saved data will be overwritten.

Let's say that you'd like to append your newDailyUsers file each day by scheduling this search to run every 24 hours:

```
| parse "name=*" as name
| parse "action=*" as action
| parse "date=*" as date
| where action="sign-up"
| first(date) as date, first(action) as action by name
| save append myFolder/mySubFolder/newDailyUsers
```

Each day the query runs the above data is appended to the newDailyUsers file.

You can also append data to a saved file from different queries. For example, say we have two sources, "bill" that includes billing information, and "config" that contains account information, and we'd like to be able to search for some values from each source. These searches would create a table with information from both sources:

```
source=bill | parse "user_id=*" as name
| parse "user_email=*" as email
| save myFolder/mySubFolder/NameEmailMapping

source=config | parse "_user=[*]" as name
| parse "contact_info=[*]" as email
| save append myFolder/mySubFolder/NameEmailMapping
```

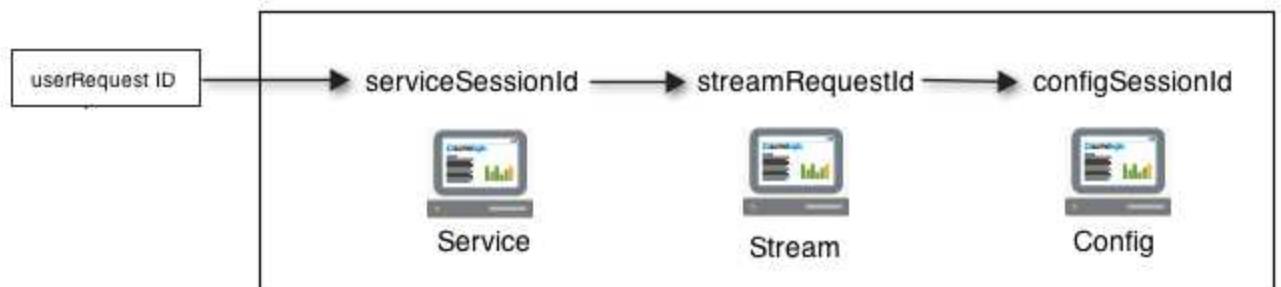
Using the Sessionize Operator

The **Sessionize** operator allows you to use an extracted value from one log message (generated from one system) to find correlating values in log messages from other systems. After you run Sessionize, these related events are displayed on the same page. The thread of logs woven together is called a **session**.



Depending on your use case, you could also use the Join operator, which may be more appropriate and easier to use.

For example, let's say we have the value of a userRequestId, which entered a distributed system; the request goes through systems named Service, Stream, and Config:



Each system generated log messages, so we know that at some point a failure occurred. We know the `userRequestId` value from the log files from the Service machine, and we know the `serviceSessionId`, `streamRequestId`, and `configSessionId`. Using `Sessionize`, we can weave together these disparate logs to identify where the failure occurred.

Rules:

- `Sessionize` operator is followed by more than one anchor expression.
- Each anchor expression can be used to extract one or more variables from a matching log.
- You can use the extracted variable to join with a second log message containing that variable using a `$variableName` notation.

Syntax Example:

```
1 (SearchServiceImpl Creating Query) or (Stream SessionId using searchSessionId) o
2 | sessionize "session: '*', streamSessionId: '*'" as (serviceSessionId, streamSe
3 "Stream SessionId=$streamSessionId using searchSessionId=* and rawSessionId=" a
4 "Started search with sessionId: $searchSessionId, customerId: *, query: *" as (c
```

1. Specify the search conditions that correlate three types of logs (not strictly required, but recommended).
2. Extract `serviceSessionId` and `streamSessionId` from the first log type.
3. Join with the second log type using `serviceSessionId`, and use that ID to extract `searchSessionId` and `rawSessionId`.
4. Join with the third log type using `searchSessionId` extracted in line three.

The above example query would produce these results:

Messages								
Page: 1 of 1								
#	Time	customerId	query	rawsessionId	searchsessionId	servicesessionid	streamsessionid	Message
1	05/30/2012 21:44:36.892	5	error	951E35CABAC8884D	5FF494E804E4F425	5CAA2089ADDE52A5	92D6FD70915C0C6B	2012-05-30 21:44:36,892 -0700 INFO [H [logger=scala.search.SearchQueryHandle [auth=User:david@demo.com:000000000000 [web_session=sqs7keby...] [session=5FF [call=InboundSearchProtocol.startSearch sessionId: 5FF494E804E4F425, customerI Host: prod-ftsearch-5 Name: /usr/sumo/search-1
2	05/30/2012 21:44:33.533	5	error	951E35CABAC8884D	5FF494E804E4F425	5CAA2089ADDE52A5	92D6FD70915C0C6B	2012-05-30 21:44:33,533 -0700 INFO [H [logger=stream.scala.session.StreamQue (group:HornetQ-client-global-threads-1 [auth=User:david@demo.com:000000000000 [web_session=sqs7keby...] [call=Inboun SessionId=92D6FD70915C0C6B using searc rawSessionId=951E35CABAC8884D Host: prod-ftsearch-2 Name: /usr/sumo/stream-1
3	05/30/2012 21:44:33.425	5	error	951E35CABAC8884D	5FF494E804E4F425	5CAA2089ADDE52A5	92D6FD70915C0C6B	2012-05-30 21:44:33,425 -0700 INFO [H [logger=service.endpoint.search.v1.imp [auth=User:david@demo.com:000000000000 [web_session=sqs7keby...] Creating que May 30 21:44:35 PDT 2012', timeZone: ' '5CAA2089ADDE52A5', streamSessionId: ' '92D6FD70915C0C6B', fromSavedSearch: 'f Host: prod-frontend-3 Name: /usr/sumo/service-1



After using the Trace operator to find related sessions, you can use the Sessionize operator to refine the results.

Smooth Operator

The **smooth operator** calculates the rolling (or moving) average of a field, measuring the average of a value to "smooth" random variation. Smooth operator reveals trends in the data set you include in a query.

Within a query that contains a smooth operator you'll choose a window (described as `window_length` in syntax below); the average of the values within the window creates a data point. The default window length is 10, but you can choose any number in your query.

Adding a group by function to a smooth operator query produces a running average within each group (with data from each group calculated separately).

Syntax:

- ... smooth field [, window_length]

Rules:

- An alias for smooth is optional. When an alias is not provided, `_smooth` is the default alias.
- Specified fields must contain numeric values.
- To add a query that includes a smooth operator to a Dashboard, you must add a group by function before the smooth operator.

Examples:

Smooth the difference of a quantity between time points. Using smooth with timeslice, you can run a query similar to:

```
* | parse "bytes transmitted: *" as bytes | timeslice 1m | sum(bytes) as bytes by _timeslice | sort _timeslice | smooth bytes, 5
```

that produces results like:

Messages		Aggregates	
		Page: 1 of 1	
#	Time	bytes	_smooth
1	10-25-2012 08:56:00	4,196,278	4,196,278
2	10-25-2012 08:55:00	10,514,578	7,355,428
3	10-25-2012 08:54:00	11,210,450	8,640,435.33333
4	10-25-2012 08:53:00	8,003,994	8,481,325
5	10-25-2012 08:52:00	14,013,801	9,587,820.20000
6	10-25-2012 08:51:00	7,263,656	10,201,295.80000
7	10-25-2012 08:50:00	7,576,588	9,613,697.80000
8	10-25-2012 08:49:00	12,061,436	9,783,895
9	10-25-2012 08:48:00	9,212,021	10,025,500.40000
10	10-25-2012 08:47:00	7,533,306	8,729,401.40000
11	10-25-2012 08:46:00	7,485,960	8,773,862.20000
12	10-25-2012 08:45:00	25,894,866	12,437,517.80000
13	10-25-2012 08:44:00	17,778,891	13,581,008.80000
14	10-25-2012 08:43:00	18,216,019	15,381,808.40000
15	10-25-2012 08:42:00	6,270,559	15,129,259

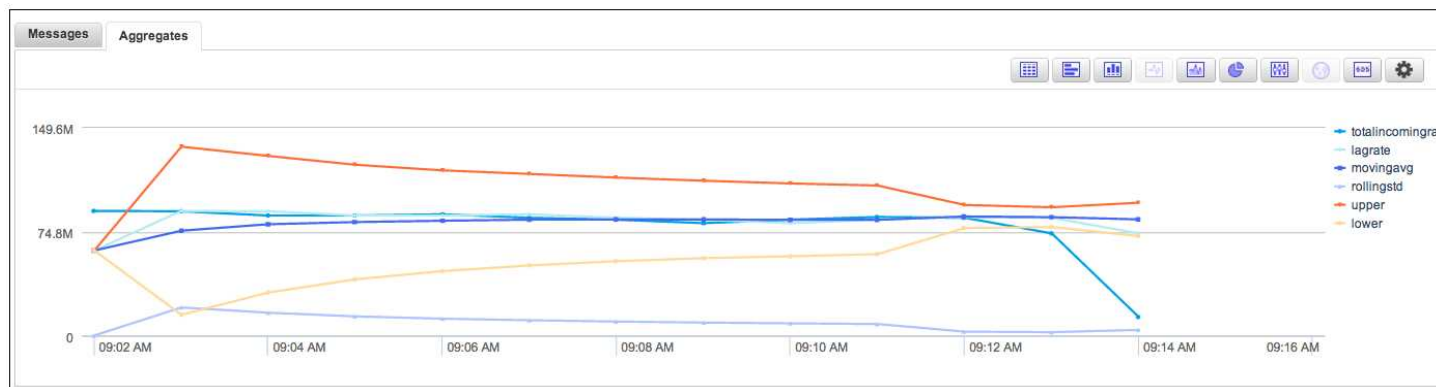
Use backshift with smooth and rollingstd to view the averages of incoming bytes.

Running a query like:

...

```
| timeslice by 1m| avg(oneMinuteRate) as avgRateByHost by _sourcehost,_timeslice
| sum(avgratebyhost) as totalIncomingRate by _timeslice
| sort + _timeslice
| backshift totalIncomingRate, 1 as lagRate
| smooth lagRate,10 as movingAvg
| rollingstd lagRate,10 as rollingStd
| movingAvg + (3 * rollingStd) as upper
| movingAvg - (3 * rollingStd) as lower
```

produces results similar to:



Sort Operator

The **sort** operator orders aggregated search results. The default sort order is descending.

Then you can use the **top** or **limit** operators to reduce the number of sorted results returned.

Syntax:

- ... | sort by fieldname (displays results as descending, by default)
- ... | sort by +fieldname (displays results as ascending)
- ... | sort by fieldname asc (displays results as ascending)
- ... | sort by fieldA, fieldB
- ... | top # fieldname by group_by_function

Rules:

- Default sort order is descending.
- To reverse the sort order to ascending, type a plus sign (+) before the field name you are sorting by. Alternatively, you can type **asc** after the field name.
- To numerically sort, first cast the field to a number. (Otherwise, the sort will be ordered as a text field.)

Examples:

- status AND down | extract "user=(?<user>.*?):" | count (*) group by user | sort by _count
- ... | count user | top 2 user by _count
- ... | count user | sort by _count asc

Top 10 pages by page hits.

In this example, you can count page hits by the source and host, then sort by page hits, and limit the results to the top 10.

```
_source*
| count as page_hits by _sourceHost
| sort by page_hits
| limit 10
```

which provides results like:

#	_sourcehost	page_hits
1	ip-10-165-35-10	50,821
2	ip-10-154-190-1	29,955
3	ip-10-244-50-11	25,328
4	ip-10-242-233-1	19,126
5	54.224.208.15	11,997
6	ip-10-155-148-23	10,253
7	poseidon.sumolab.org	8,615
8	ip-10-185-2-1	7,941
9	Sumo Cloud	7,288
10	db.stocktrader.com	2,693

For more information, see [Top Operator](#) or [Limit Operator](#).

Split Operator

The Split operator allows you to split strings into multiple strings, and parse delimited log entries, such as space-delimited formats.



To parse log entries from CSV files, you can use the simpler [CSV operator](#).

Syntax:

Extract fields using index:

- `split fieldName extract 1 as A, 2 as B, 5 as E, 6 as F`

Extract fields using position:

- `split fieldName extract A, B, _, _, E, F`

Mix positional and index-based:

- `split fieldName extract A, B, 5 as E, F`

Extract from an existing field:

- `parse "start*end" as fieldName | split fieldName extract 1 as A, 2 as B, 5 as E, 6 as F`

Specify a delimiter, escape, and quote character:

- `split fieldName escape='\', delim=':', quote='"' extract A, B, _, _, E, F`

Rules:

- By default, the Split operator uses a comma (,) for a delimiter, backslash (\) for an escape character, and (") quote for a quote character, though you can define your own if you like.
- If you define your own escape, delimiter, and quote characters, they must all be different.
- A field name is always required.

Examples:

Parsing a colon delimited file.

For example, if you had a file with the following colon delimited log message:

```
[05/09/2014 09:39:990] INFO little@sumologic.com:ABCD00001239:EFGH1234509:
"Upload Complete - Your message has been uploaded successfully."
```

You could parse the fields using the following query:

```
_sourceCategory=colon
| parse "]" * "*" as log_level, split_field
| split split_field delim=":" extract 1 as user, 2 as account_id, 3 as session_id, 4 as result
```

which produces results such as:

#	Time	log_level	split_field	user	account_id	session_id	result	Message
1	05/09/2014 11:00:11.242	INFO	little@sumologic.com:ABCD00001239:EFGH1234509:"Upload Complete - Your message has been uploaded successfully."	little@sumologic.com	ABCD00001239	EFGH1234509	Upload Complete - Your message has been uploaded successfully.	[05/09/2014 09:39:990] INFO little@sumologic.com:ABCD00001239:EFGH1234509:"Upload Complete - Your message has been uploaded successfully." Host: 12.177.21.34 ▼ Name: Http Input ▼ Category: colon ▼
2	05/09/2014 11:00:11.242	INFO	sam@sumologic.com:ABCD00001237:EFGH1234569:"Upload Complete - Your message has been uploaded successfully."	sam@sumologic.com	ABCD00001237	EFGH1234569	Upload Complete - Your message has been uploaded successfully.	[05/09/2014 09:38:178] INFO sam@sumologic.com:ABCD00001237:EFGH1234569:"Upload Complete - Your message has been uploaded successfully." Host: 12.177.21.34 ▼ Name: Http Input ▼ Category: colon ▼
3	05/09/2014 11:00:11.242	INFO	fred@sumologic.com:ABCD00001244:EFGH1234579:"Upload Complete - Your message has been uploaded successfully."	fred@sumologic.com	ABCD00001244	EFGH1234579	Upload Complete - Your message has been uploaded successfully.	[05/09/2014 09:37:156] INFO fred@sumologic.com:ABCD00001244:EFGH1234579:"Upload Complete - Your message has been uploaded successfully." Host: 12.177.21.34 ▼ Name: Http Input ▼ Category: colon ▼

In another example, you could use the following query:

```
_sourceCategory=colon
| split _raw delim=":" extract 1 as user2, 2 as id, 3 as name
```

which provides results like:

#	Time	user2	id	name	Message
1	05/08/2014 17:01:33.735	user10	10	Franklin Trevaleon	user10:10:"Franklin Trevaleon" Host: 108.248.126.124 ▼ Name: Http Input ▼ Category: colon ▼
2	05/08/2014 17:01:33.735	user9	9	Napoleon Trois	user9:9:"Napoleon Trois" Host: 108.248.126.124 ▼ Name: Http Input ▼ Category: colon ▼
3	05/08/2014 17:01:33.735	user8	8	Domigo Montoya	user8:8:"Domigo Montoya" Host: 108.248.126.124 ▼ Name: Http Input ▼ Category: colon ▼

Parsing a CSV file.

Use the following query to extract comma delimited fields as specified:

```
_sourceCategory=csv
| split _raw delim=',' extract 1 as user2, 2 as id, 3 as name
```

which produces results such as:

#	Time	user2	id	name	Message
1	05/08/2014 16:59:56.761	user10	10	Franklin Trevaleon	user10,10,"Franklin Trevaleon" Host: 108.248.126.124 ▼ Name: Http Input ▼ Category: csv ▼
2	05/08/2014 16:59:56.761	user9	9	Napoleon Trois	user9,9,"Napoleon Trois" Host: 108.248.126.124 ▼ Name: Http Input ▼ Category: csv ▼
3	05/08/2014 16:59:56.761	user8	8	Domigo Montoya	user8,8,"Domigo Montoya" Host: 108.248.126.124 ▼ Name: Http Input ▼ Category: csv ▼

Summarize Operator

The **Summarize** operator's algorithm uses fuzzy logic to cluster messages together based on string and pattern similarity. You can use the summarize operator to quickly assess activity patterns for things like a range of devices or traffic on a website. Focus the summarize algorithm on an area of interest by defining that area in the keyword expression. For information on how to interpret and influence the outcome of Summarize results, see [Detecting Patterns with Summarize](#) and [Influencing the Summarize Outcome](#).



The **LogReduce** button in the **Messages** tab allows you to apply the **Summarize** operator automatically to the results of a non-aggregated search.

Syntax:

- keyword expression | summarize

Rules:

- Cannot be used with group-by functions such as "count by fieldname"

Examples:

- * | summarize
- _sourceCategory=firewall | summarize
- error or exception | summarize
- _sourceCategory="Western Region" | summarize

See Also:

[Influencing the Summarize Outcome](#)

[Summarize Save as Baseline](#) (Sumo Logic Lab)

[Searching with LogReduce](#)

Detecting patterns with summarize

The **Summarize** algorithm uses fuzzy logic and soft matching to group messages with similar structures and common repeated text strings into **signatures**, providing a quick investigative view, or snapshot, for the keywords or time range provided.

The **Signature** column in the **Summarize** tab displays **signatures**. A signature is basically a reflection of the logs grouped by summarize—not all logs grouped in a signature will exactly match it. Within a signature, fields that vary are displayed with wildcard placeholders (*****) while other fields, such as timestamp (and some URLs) are ignored and replaced with placeholder variables such as \$DATE and \$URL.

You can refine the results of the summarize algorithm to make the outcome more generic or more specific. See [Influencing the Summarize Outcome](#) for more information.

Will my summarize search results match my keyword search results?

Generally speaking, no. Summarize is intended to be a jumping-off point for your analysis. Unlike a keyword search, where you're looking for data related to, say, a specific Source or an error message, summarize returns signatures that contain messages that *may* be of interest to you using fuzzy logic. If you're not happy with a signature, you can **teach** summarize how you'd like the results to be made more specific. Don't think of a signature as an example of what logs are grouped under it; instead think of a signature as a reflection of what summarize thinks you'll find interesting if that signature catches your eye. Once you begin digging in to summarize results, you'll then want to structure a keyword query that delivers precise results.

Running a Summarize query

When you run a Summarize query, you can first filter results with a simple string or metadata expression [or you can just type a wildcard (*)]. Specify a reasonable time period, service, or geographic region. Follow your keyword expression with the Summarize operator to group the resulting logs into meaningful groups of messages called **signatures**. When running a Summarize query, you will often see signatures change as the algorithm sorts through the resulting data and works to determine the best signature assignments for messages.



Summarize cannot be used with [group-by functions](#) such as "count by fieldname".

To run a Summarize query:

1. In the search query field, enter a keyword string or a metadata tag (for example, `_sourceCategory="Western Region"`) to initially filter messages to some category [or you can just type a wildcard (*)].
2. Then type a pipe symbol (|) and the **summarize** operator. For example, to summarize messages for your "CustomerAccounts" module, type:

CustomerAccounts | summarize

3. Press enter or click **Start**. Results appear in the **Summarize** tab. Do any of the following:
 - Click the **Messages** tab to see the individual messages for all signatures combined.
 - To see the messages grouped in a signature, select the check box for the signature, and then click **View Details**. A new Search tab opens with the messages displayed. You can check more than one box to see the results in time <<>> order in the new **Search** tab.
 - To export the results, click the **Export** icon. Then click **Download** to save the file to your computer.
 - To save the summarize query as a Summarize Baseline, click the **Save Baseline** icon. Enter a Name for the baseline and then click **Save**.

The screenshot shows the Summarize tab interface. At the top, there are tabs for 'Messages' and 'Summarize'. Below the tabs is a pagination bar showing 'Page: 1 of 2'. The main table has columns: #, Select, Count, Relevance, Actions, and Signature. The first row is highlighted with a red box labeled 'A' pointing to the 'Actions' column, which contains icons for Promote, Demote, and Split. The second row is highlighted with a red box labeled 'B' pointing to the 'Select' column, which contains a checkbox. The table lists several signatures with their respective counts and relevance scores.

#	Select	Count	Relevance	Actions	Signature
1	<input type="checkbox"/>	2	9.53		[hornetq-failure-check-thread; threadId: *3] \$DATE WARNING [org.hor detected: Did not receive data from /1*****. It is likely the clie You also might have configured connection-ttl and client-failure-ch
2	<input type="checkbox"/>	450	5		\$DATE [metrics-log-reporter] INFO com.sumologic.util.scala.Metrics fiveMinuteRate=*.00, oneMinuteRate=*.00, meanRate=0.*
3	<input type="checkbox"/>	184	5		\$DATE WARN [hostId=stag-config-1] [thread=*****@ntp-*****-13*] (com.sumologic.interchange.config.v1.Contigerrors
4	<input type="checkbox"/>	108	5		\$DATE ERROR [hostId=stag-cq-1] [module=CQ] [logger=scala.c*****tio ***** 000000000000 ***** n: com.sumologic.scala.config.dm.org.Gro
5	<input type="checkbox"/>	88	5		\$DATE INFO [hostId=stag-c***] [module=**] [logger=com.sumologic. count=1, fifteenMinuteRate=0.00, fiveMinuteRate=0.00, oneMinuteRate
6	<input type="checkbox"/>	85	5		\$DATE INFO [hostId=stag-ftsearch-1][logger=com.sumologic.s*****)] fifteenMinuteRate=****, fiveMinuteRate=****, oneMinuteRate=****,
7	<input type="checkbox"/>	72	5		\$DATE ERROR [hostId=stag-frontend-1] [module=RECEIVER] [web_session [auth=Collector:*****:0000000000 *****:0000000000 *****:false] [MAP is not supported

- A. Promote, Demote, and Split icons.
- B. Undo and Redo icons.
- C. Click to view messages for the selected signature.
- D. Click to save the query as a Baseline.
- E. Click to download the Summarize report.

Investigating the Others signature

Messages that Sumo Logic cannot readily group are separated into a distinct signature called **Others**. These signatures might contain simple miscellaneous messages that are not important, or it might show some anomalous messages that are meaningful.

To investigate the messages in the **Others** signature:

1. Select the check box and click **View Details**.
2. Sumo Logic runs the Summarize algorithm on the signature, and then displays the resulting sub-signatures.



Visit [Sumo Logic Labs](#) as a logged-in user to try out the new beta feature for Summarize: Run a Delta Against a Baseline. Save a named baseline, and then run a delta against the baseline to see changes in your log patterns over time.

Viewing Summarize Details

From the **Summarize** tab, you can view logs grouped together in a signature to view the raw log data by clicking the number in the **Count** column:

Messages		Summarize									
		Page: 1 of 23				View Details					
6	<input type="checkbox"/>	16	9.53				\$DATE - INFO [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:2181:NIOServerCxn@1435] - Closed socket connection for client /10.1 ***** which had sessionid 0x13 *****				
7	<input type="checkbox"/>	15	9.53				\$DATE - INFO [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:2181:NIOServerCxn@777] Client attempting to establish new session at /10.1 *****				
8	<input type="checkbox"/>	2	9.53				\$DATE INFO [hostId=nite- ****-1] [module=****] [logger=org.springframework.beans.factory.xml.XmlBeanDefinitionReader] [thread=****] Loading XML bean definitions from **** resource [****t.xml]*				
9	<input type="checkbox"/>	5,350	5				\$DATE INFO [hostId=nite-cass-meta-1] [module=META] [logger=scala.config.protocol.handler.ConfigCollectorProtocolHandler] [thread=NervHeartbeatMonitorLeader] Updated state for 0000000000 *****7. alive:false, lastSeen:*** Jan***** PST 2013				
10	<input type="checkbox"/>	2,817	5				\$DATE [Thread-* (group:HornetQ-client-global-threads-17*****)] INFO com.sumologic.scala.collector.CommonsHTTPSender - Publishing message piles: '1', messages: '***', bytes: '*****', encoded: '*****', threshold: 'false', compressed: '****', by transmitter: '0'				







Viewing data in this manner doesn't supply feedback to the Summarize operator; it simply launches a new query that allows you to view the logs grouped under that signature. As you're reviewing the data, you may decide to edit the signature or split the signature, as you see fit.

The Summarize details query uses the line number of the signature to return these results in a new search tab.

Influencing the Summarize Outcome

The algorithm used for the **Summarize** operator uses fuzzy logic and soft matching to group messages with similar structures and common repeated text strings into **signatures**, providing a quick investigative view, or snapshot, for the keywords or time range provided. Summarized data is based on the data available to the algorithm during the time range of your search. In some cases, this data sampling produces results that don't meet your needs. You can influence the algorithm by editing a signature to make the results more general, or see more granular results by splitting a signature. In addition you can promote or demote a signature to help Sumo Logic understand what data is the most relevant to you.

The following icons allow you to change the results of a summarize report:

Icon	Action
	Promote a signature to the top position of the Summarize tab.
	Demote a signature to move it to the bottom of the last page of the Summarize tab.
	<u>Split</u> a signature into multiple signature.
	Edit a signature.
	Undo the last action or step back through the history of changes.
	Redo the last action. Repeat to redo the history of undos.

Promoting or Demoting a Summarize Signature

When viewing **Summarize** results, you can **promote** a signature to indicate to Sumo Logic that the data included in the signature is exceptionally relevant to you. This feedback is taken into consideration when you run Summarize the next time. If you click the thumbs down button to **demote** a signature, Sumo Logic also learns that the data contained in the signature isn't highly relevant; in future searches signatures containing that data won't appear in the top of the results tab.

If, however, you run a saved search containing Summarize, promoting or demoting a signature will apply only to the results of the single search, and won't be applied to searches you run ad-hoc. In other words, the promotion or demotion is retained only in the context of the saved search results you're viewing.

To promote a signature:

- Click the **thumbs up** icon to move the signature to the top position in the **Summarize** tab.

Messages

Summarize

⏮

⏪

Page: 1

of 1






⏩

⏭

↺

↻

View Details

#	Select	Count	Relevance ▼	Actions	Signature
1	<input type="checkbox"/>	4	10	  	[New I/O server worker #2-*; threadId:4**] Reattach request from / *****.174.225:***** failed as there is no confirmationWindowSize
2	<input type="checkbox"/>	22	9.53	  	[hornetq-failure-check-thread; threadId:**] \$DATE WARNING [org.*** detected; Did not receive data from /1*****, [code=*****
3	<input type="checkbox"/>	8	5	  	[hornetq-failure-check-thread; threadId:**] \$DATE WARNING [*****

To demote a signature:

- Click the thumbs down icon to move the signature to the bottom of the last page in the Summarize tab.

Messages

Summarize

⏮

⏪

Page: 1 of 1

⏩

⏭

↺

↻

View Details






















#	Select	Count	Relevance ▼	Actions	Signature
1	<input type="checkbox"/>	22	8.42	<div><div>👍</div><div>👎</div><div>🔄</div></div>	[hornetq-failure-check-thread; threadId:**] Did not receive data from / Connection failure has been detected server and client has failed code=3
2	<input type="checkbox"/>	8	5	<div><div>👍</div><div>👎</div><div>🔄</div></div>	[hornetq-failure-check-thread; threadId:**] \$DATE WARNING ***** Cleared up resources for session *****11e2-*****
3	<input type="checkbox"/>	4	0	<div><div>👍</div><div>👎</div><div>🔄</div></div>	[New I/O server worker #2-*; threadId:4**] Reattach request from ***.188.174.225:***** failed as there is no confirmationWindowSize conf

Splitting a signature




























If you'd like to see more granular results, you can **split** a signature. When you split a signature, you'll notice that fewer wildcard asterisks will appear; instead specific values are included in the signatures. Even though the data is more specific, the results after splitting a signature will still be fuzzy because the summarize algorithm bases results only on the window of time you've run the search against.

After you split a signature, the position of the signatures may move (one may even move to another page of results). Each line is still highlighted in yellow so you can easily identify them.

For example, we've selected a signature to split. We know that the hostId shouldn't be generic; by splitting the signature we should get more specific results.

Messages		Summarize			
		Page: 1 of 3			
#	Select	Count	Actions	Signature	
1	<input type="checkbox"/>	7,549	  	\$DATE INFO [hostId=long-frontend-*] [module=RECEIVER] [logger=util.scala.MetricsReporter] [thread=metr...	
2	<input type="checkbox"/>	454	  	\$DATE [metrics-log-reporter] INFO com.sumologic.util.scala.MetricsReporter - com.sumologic.scala.collec...	
3	<input type="checkbox"/>	443	  	\$DATE INFO [hostId=long-cq-1] [module=cq] [logger=util.scala.MetricsReporter] [thread=metrics-log-repor...	
4	<input type="checkbox"/>	204	  	\$DATE INFO [hostId=long-*****-1] [module=***] [logger=util.scala.MetricsReporter] [thread=metrics-...	
5	<input type="checkbox"/>	94	  	\$DATE ERROR [hostId=long-cass-meta-1] [module=NERV] [logger=nerv.EKG] [thread=NervHeartbeatMonitorLeader...	
6	<input checked="" type="checkbox"/>	84	  	\$DATE INFO [hostId=long-f*****] [module=*****] [logger=util.scala.MetricsReporter] [thread=metri...	
7	<input type="checkbox"/>	52	  	\$DATE INFO [hostId=long-katta-13] [module=KATTA] [logger=net.sf.katta.lib.lucene.LuceneServer] [thread=...	

After splitting, you'll see that each signature has specific data:

Messages		Summarize			
		Page: 1 of 3			
6	<input type="checkbox"/>	52	  	\$DATE INFO [hostId=long-katta-13] [module=KATTA] [logger=net.sf.katta.lib.lucene.LuceneServer] [thread=IPC...	
7	<input type="checkbox"/>	42	  	\$DATE INFO [hostId=long-ftsearch-1] [module=search] [logger=util.scala.MetricsReporter] [thread=metrics-log...	
8	<input type="checkbox"/>	26	  	\$DATE ERROR [hostId=long-config-1] [module=CONFIG] [logger=scala.concierge.schedule.ScheduleManager] [thread...	
9	<input type="checkbox"/>	22	  	\$DATE DEBUG [hostId=long-*****] [module=CQ] [logger=scala.cq.worker.DefaultContinuousQueryEngineManager] [th...	
10	<input type="checkbox"/>	14	  	\$DATE INFO [hostId=long-config-1] [module=config] [logger=util.scala.MetricsReporter] [thread=metrics-log-r...	
11	<input type="checkbox"/>	14	  	\$DATE INFO [hostId=long-config-1] [module=config] [logger=util.scala.MetricsReporter] [thread=metrics-log-r...	
12	<input type="checkbox"/>	14	  	\$DATE INFO [hostId=long-config-1] [module=config] [logger=util.scala.MetricsReporter] [thread=metrics-log-r...	
13	<input type="checkbox"/>	14	  	\$DATE INFO [hostId=long-config-1] [module=config] [logger=util.scala.MetricsReporter] [thread=metrics-log-r...	
14	<input type="checkbox"/>	13	  	\$DATE ERROR [hostId=long-frontend-1] [module=SERVICE] [logger=scala.config.graphORM.Transaction\$] [thread=bt...	

To split a signature:

- Click the Split icon next to the signature you'd like to split.

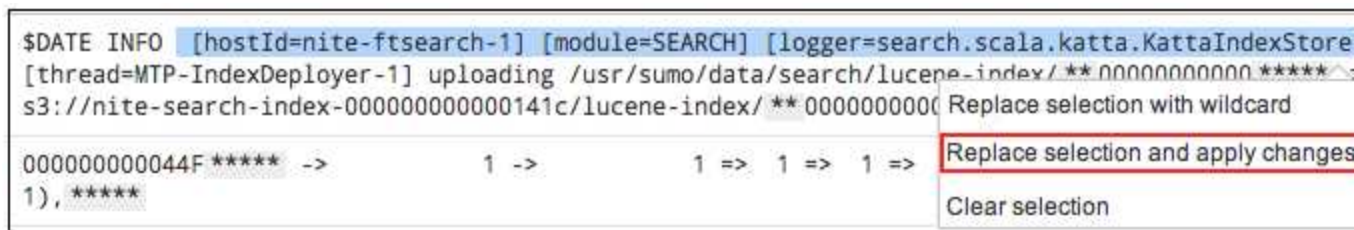
Note that the previous line 7 (from Figure 1, above) was combined with the line above it, where the sessionID value was already being treated as a wildcard. This means that the line has jumped up to number 5 (in Figure 2), and that there are more results. If more specific results are needed, we could split the signature to break them out once again.



When in doubt, replace fewer characters with a wildcard to avoid over-generalizing the signature.

To edit a signature:

1. Select text in the signature you'd like to edit. Then click **Replace selection with wildcard**.

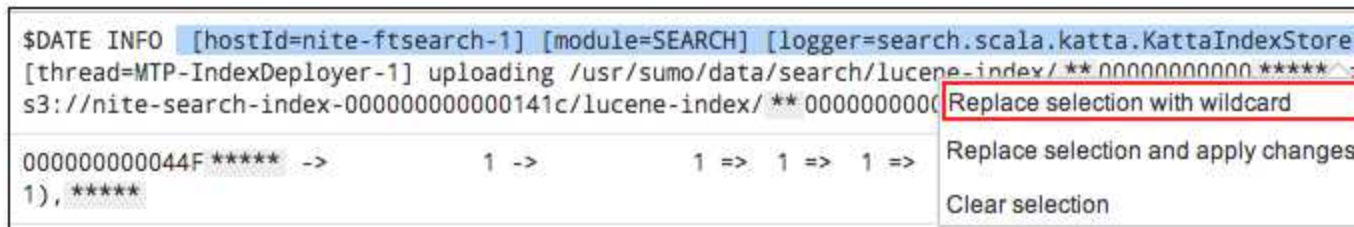


2. (Optional) If you're not happy with the results, while the line is still selected, click **Clear selection**, or click the **Undo** icon.

To make multiple edits to a single signature:

If there are two strings you'd like to replace with a wildcard, you can make the changes, then apply them at the same time. You'll have to click the **Apply Changes** button to save the wildcards.

1. Select some of the text you'd like to replace with a wildcard, and click **Replace selection with wildcard**. Then select the next portion of text you'd like to replace, and click **Replace selection with wildcard**.



2. Click **Apply Changes**.



Understanding the Summarize Relevance column

The **Summarize Relevance** column displays a numerical score for a signature, predicting which signatures could be most meaningful to a user. The Relevance value is computed using your history of [feedback](#) (**Thumbs Up** and **Thumbs Down**) and the instances when you've chosen to view the details of signatures.

Summarize uses the similarity of signature content (the words in a signature) to predict relevance for signatures, even if a signature hasn't yet been promoted or demoted a specific signature. For example, if a user has promoted a number of signatures that contain the word "database" then new signatures containing "database" will be scored higher.

Messages

Summarize

⏮

⏪

Page: 1 of 3










⏩

⏭

↺

↻

View Details

#	Select Count ▼	Relevance	Actions	Signature
1	<input type="checkbox"/> 3,956	8.3	  	\$DATE WARN [hostId=*****] [module=*****] [logger=scala.config.*****] [thread=*****] [auth=SystemUser:*****User:000000000000 *****:FFFFFFFFFFFFD66:false] Permissions errors have b
2	<input type="checkbox"/> 2,392	5	  	\$DATE INFO [hostId=l*****] [module=**] [logger=***** DefaultEngineManager] [thread=DefinitionSource(reload CQs)] Current running engine: DefinitionAndCustomer([definition ProtocolContinuousQueryId(*****), userId: *****, query:***** ***** as **** *****by ****
3	<input type="checkbox"/> 2,014	3.2	  	\$DATE INFO [hostId=config-1] [module=CONFIG] [logger=***** server.ConfigServer] [thread=Thread- ***** (group:*****.global-threads- *****)]

What do the Relevance values mean?

Relevance ranks signatures on a scale of 0 to 10. Values of 0, 5, and 10 have specific meanings; values falling between these numbers suggest that a signature itself has not been explicitly promoted or demoted, but that it contains terms that have received previous feedback (either positive or negative).

10	Assigned to signatures explicitly given a Thumbs Up .
5	The default value for signatures that have no content in common with logs that have received feedback (neither positive or negative). Think of 5 as a neutral relevance value for signatures that contain words that Summarize hasn't learned how to rank.
0	Assigned to signatures that were explicitly given a Thumbs Down .

Changing the display of Summarize results

Summarize results are displayed in descending order of the Relevance value by default. You can view results by Count if you prefer.

To re-sort Summarize results by Count:

- If you prefer to see data sorted by the value in the **Count** column, click the **Count** column header to re-sort by Count.

Using Summarize Delta

With **Summarize Delta**, you can save a baseline summarize pattern, and then later run a delta against the baseline. Sumo Logic will report the variance giving you insight into the shifts in patterns over time.

The baseline saves as a summarize pattern independent of the time range or the keywords used for the baseline. You can run a delta against the baseline even if you are using different keywords or a different time range.

To run a Summarize Delta:

1. Save a baseline using a path-like naming structure. For example, the query might look like:

```
error or failure | summarize save /your_category/errors_baseline
```



Make a note of the baseline name since there is currently no UI to save this for you.

2. When you're ready, run a **delta** against the saved baseline, either using the same keyword expression, or a new keyword expression. So the delta query might look like:

```
exceptions | summarize delta /your_category/errors_baseline
```

Understanding Summarize Delta results

In the delta results, you might see that some clusters are "gone" and some are "new," especially if you have used a keyword expression that is different from the baseline. You will also see some new information displayed in the **Count** column for results in the Web Application. These include:

- **Delta Percentage.** The delta percentage is the straight percentage change in the number of messages divided by the historical count percentage, which is the historical count change over time since the initial count results for this cluster.
- **New.** When the word **New** appears, the cluster did not exist in the original baseline.
- **Gone.** When the word **Gone** appears, a previous cluster did not return in the delta summarize.

Options

There are a few options to use with the Summarize Delta feature including **update**, **purge**, and **nopurge**.

update. You can update the saved baseline at any time using the update option. When you use the update option, empty clusters older than seven days are purged and the new summarize results are saved as the baseline. The syntax is as follows:

```
<keyword expression> | summarize delta path/to/named_baseline update
```

purge. You can combine update and purge to overwrite the baseline and remove empty clusters at the same time. With the purge option, you can set a specific time period other than seven days to purge empty clusters. The syntax for the time expression is relative to now. In this example, we are updating the baseline and purging clusters older than three days:

```
<keyword expression> | summarize delta path/to/named_baseline update purge 3d
```

nopurge. Additionally, you have the option to keep empty clusters indefinitely using the nopurge option:

```
<keyword expression> | summarize delta path/to/named_baseline update nopurge
```

Extending Summarize Delta with the Where Operator

When you use the Summarize operator, new fields are created that you can use to focus your delta results. For example, you can run the query:

```
<keyword expression> | summarize delta path/to/named_baseline | where (_isNew)
```

The query results are constrained to new clusters only using the `_isNew` field.

In this example query, the summarize command shows only clusters that no longer include any messages:

```
<keyword expression> | summarize delta path/to/named_baseline | where _count == 0
```

Field Descriptions

The fields created with the Summarize query include the following.

Field	Description
<code>_count</code>	The number of log messages that belong to this cluster for this query.
<code>_percentage</code>	The delta percentage change between the total number of messages and the previous <code>_count</code> .
<code>_historicalCount</code>	The count of the last time this cluster had non-zero number of matching logs.
<code>_historicalPercentage</code>	<code>_percentage</code> of the last non-zero encounter.
<code>_deltaPercentage</code>	The delta percent change over time calculated as $(_percentage - _historicalPercentage) / _historicalPercentage$
<code>_anomalyScore</code>	The anomaly score gives you a sense of the weight of the delta change and is defined as $Math.abs(_deltaPercentage) \times Math.max(_percentage, _historicalPercentage)$
<code>_isNew</code>	True if the cluster is new, otherwise false.

Using summarize on JSON logs

If you're collecting JSON logs, you can use the summarize operator to analyze a single field instead of full raw messages. This avoids having summarize consider the repetitive headers and metadata in JSON logs. Make sure to choose a field that contains enough data for summarize to detect patterns; fields that output short value strings may not produce meaningful results.

All other [summarize features](#) can be used with the results of a summarize query run against a field.



Because summarize is meant to detect patterns in raw data, this particular application of summarize should only be used for JSON logs.

To run summarize on JSON logs:

1. Choose the field you'll use to run summarize.
2. In the **Search** tab of the Sumo Logic Web Application, run a search using the following syntax:

```
* | parse "[pattern]" as [fieldname] | summarize field=[fieldname]
```

3. Hit enter or click **Start**. Results appear in the **Summarize** tab. Do any of the following:
 - Click the **Messages** tab to see the individual messages for all signatures combined.
 - To see the messages grouped in a signature, select the check box for the signature, and then click **View Details**. A new **Search** tab opens with the messages displayed. You can check more than one box to see the results in time order in the new **Search** tab.
 - To export the results, click the **Export** icon. Then click **Download** to save the file to your computer.
 - To save the summarize query as a Summarize Baseline, click the **Save Baseline** icon. Enter a Name for the baseline and then click **Save**.



If you run a summarize delta against the baseline, it's a good idea to specify the same field to avoid running the delta against all the fields in your logs.

Timeslice Operator

The **timeslice** operator segregates data by time period, so you can create bucketed results based on a fixed width in time, for example, five minute periods. Timeslice also supports bucketing by a fixed number of buckets across the search results, for example, 150 buckets over the last 60 minutes. An alias for the timeslice field is optional. When an alias is not provided, a default **_timeslice** field is created.

Syntax:

- ... | timeslice by <time_period> | aggregating_operator _timeslice
- ... | timeslice # buckets as <fieldname> | aggregating_operator <fieldname>

Rules:

- An alias for the timeslice field is optional. When an alias is not provided, a default **_timeslice** field is created.
- Must be used with an aggregating operator such as count by or group by.
- Results for the timeslice column are displayed in milliseconds.

Examples:

- Timeslice by 5m
Fixed-size buckets at 5 minutes. The output field is the default **_timeslice**.
- Timeslice by 2h as 2hrs
Fixed-size buckets that are 2 hours long. The output field name is aliased to "2hrs".
- Timeslice 2h as 2hrs
This is a shorthand of the previous example. The "by" keyword is optional.
- Timeslice 150 buckets
This means bucketing to 150 buckets over the search results.

- Timeslice by 1m as my_time_bucket_field_name
Fixed-size buckets of 1 minute each. The output field name is aliased to "my_time_bucket_field_name".

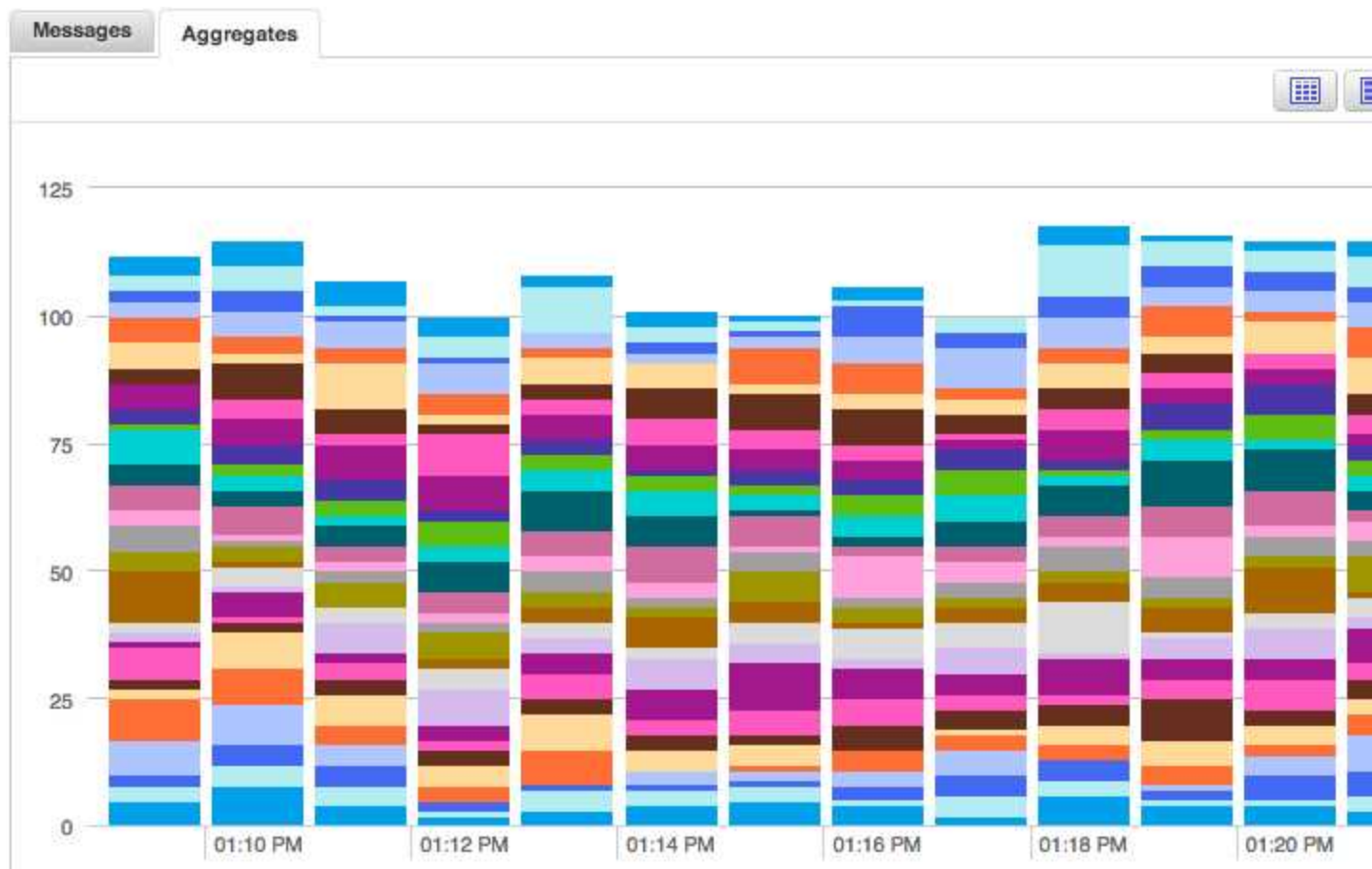
Examples in full queries:

- * | timeslice by 5m | count by _timeslice
This outputs a table in the **Aggregates** tab with columns _count and _timeslice with the timeslices spaced out by 5 minute intervals.
- * | timeslice by 5m as my_field_name_alias | count by _sourceCategory, my_field_name_alias
This outputs three columns: _count, _sourceCategory, and my_field_name_alias.
- * | timeslice 10 buckets | count by _sourceCategory, _timeslice
This outputs a table in the **Aggregates** tab with columns _count, _sourceCategory, and _timeslice with 10 rows for each _sourceCategory in that table if you have messages covering the entire search period.

The following query (also using the Transpose operator):

```
_sourceCategory=*IIS* | parse using public/iis | timeslice 1m | count  
by _timeslice, s_ip | transpose row _timeslice column s_ip
```

produces these results in the **Aggregates** tab, which you can display as a column chart:



toLowerCase and toUpperCase Operators

As the name implies, the **toLowerCase** operator takes a string and converts it to all lower case letters. The **toUpperCase** operator takes a string and converts it to all upper case letters.

These operators can be useful for normalizing source logs with inconsistent capitalization, such as Windows Event logs, or changing file names and paths for files systems that require all lower case letters. They are especially useful for queries that include conditionals and grouping, in order to reduce the number of groups in search results.

Syntax:

- ...| toLowerCase(string1) as string2
- ...| toUpperCase(string1) as string2

Rules:

- An "as field" argument is required.
- Non-string fields are not accepted.

Examples:

Using toUpperCase with a conditional operator.

Use the following query to return all the `_sourceHost` matches in upper case letters.

```
_sourceCategory=service OR _sourceCategory=search  
| toUpperCase(_sourceHost) as _sourceHost  
| where _sourceHost matches "NITE*"
```

which provides results like:

#	Time	Message
1	06/09/2014 14:30:14.401	2014-06-09 14:30:14 InMemoryIndexBuild [call=MessageProto Host: NITE-INDEX-2 ▼ hostid: nite-index-2 local
2	06/09/2014 14:30:14.380	2014-06-09 14:30:14 InMemoryIndexBuild [call=MessageProto Host: NITE-INDEX-2 ▼ hostid: nite-index-2 local

Using toUpperCase with the Count operator:

This query also returns all `_sourceHost` matches in upper case letters, using a Count operator.

```
_sourceCategory=service OR _sourceCategory=search  
| toUpperCase(_sourceHost) as _sourceHost  
| count by _sourceHost
```

which produces results like:

#	_sourcehost	_count
1	NITE-INDEX-3	6,083
2	NITE-FRONTEND-2	3,121
3	NITE-INDEX-2	63,435
4	NITE-FRONTEND-1	1,779
5	NITE-INDEX-1	4,487
6	NITE-INDEX-4	161,592

Find a user name and convert it to lowercase.

This query will search a Source Category for a user name and convert it to lowercase, no matter how the name has been input.

```
_sourceCatgeory=web  
| parse "user=* " as username  
| toLowerCase(username) as username  
| where username matches "*joe smith*"
```

Top Operator

Use the **top** operator with the sort operator, to reduce the number of sorted results returned.

Syntax:

- ... | top # fieldname by group_by_function

Examples:

List the Top 5 source categories with errors.

Use the following query to list the top 5 source categories with errors, and get their count.

```
error | top 5 _sourcecategory
```

which produces results like:

#	_sourcecategory	_count
1	esx_perf	2,024
2	CommVault	925
3	OS/Windows	607
4	OS/Linux/Security	320
5	Symantec/AntiVirus	128

You can use the following query to get the same results, but make the count explicit:

error | top 5 _sourcecategory by count

List the Top 10 source categories by message time.

This query lists the top 10 source categories by message time, without an explicit count.

error | top 10 _sourcecategory by _messagetime

which produces results like:

#	_sourcecategory	_messagetime
1	cloudfront	1,398,803,627,000
2	cloudfront	1,398,803,627,000
3	cloudfront	1,398,803,627,000
4	cloudfront	1,398,803,607,000
5	cloudfront	1,398,803,607,000
6	Symantec/AntiVirus	1,398,803,603,000
7	Symantec/AntiVirus	1,398,803,603,000
8	cloudfront	1,398,803,597,000
9	cloudfront	1,398,803,597,000
10	Symantec/AntiVirus	1,398,803,593,000

See [Sort Operator](#) for more information.

Total operator

The total operator calculates the grand total of a field and injects that value into every row.

The total operator also supports grouping rows by a set of fields.

Syntax:

- total field
The above calculates the total between consecutive values of data.
- total field [as alias] [by field1, field2, ...]
The above groups the rows of data by the set of fields specified in the by clause, and calculates the grand total within each group.

Rules:

- An alias for total is optional. If no alias is given, **_total** is used by default.
- A specified field must contain numeric values.
- If a row contains non-numeric values, that row will be skipped.

Examples:

Calculate the total. We'd like to find the total data (bytes) transmitted for a time range. Running a query similar to:

```
* | parse "bytes:*, " as data | total data as t_data
```

produces results similar to:

Messages			
#	Time	data	t_data
1	02/04/2013 09:48:24.406	1045	199,765
2	02/04/2013 09:47:41.313	1610	199,765
3	02/04/2013 09:47:41.292	1508	199,765
4	02/04/2013 09:47:40.756	1020	199,765
5	02/04/2013 09:47:21.247	1283	199,765

Calculate the running total of requests. Say we'd like to find the running total of requests from certain users.

Running a query similar to:

```
_sourceCategory=IIS (Wyatt OR Luke)  
| parse using public/iis  
| timeslice by 1m  
| count as requests by _timeslice,cs_username  
| sort by _timeslice asc, cs_username  
| total requests as running_total by cs_username
```

produces results similar to:

Imagine that an error happened at some point in the process, generating an error including "PROCESSING FAILED: webID=7F92. Starting from this information, we can use a trace operator in our query to following the chain of activity:

```
* | trace "ID=( [0-9a-fA-F] {4} )" "7F92"b | where _raw matches "**ERROR**"
```

This query tells trace how to identify the individual pieces of the chain, using the four-digit hexadecimal string following "ID=". Trace then scans incoming logs to connect the dots, building a chain based on IDs occurring together in the same log, starting from the value we supplied (7F92 in our example). So if trace observes a long, "Initiating requestID=082A for webID=7F92" it identifies the relationship between the webID we supplied with the requestID. Trace will continue to scan logs, building the chain of events. Log messages unrelated to these values are disregarded.

Tracing forward and backward in time

You can use a trace operator to trace events in the past or to track future events. In either case, a chain is built, finding links between log messages to determine activity based on whatever values you query. For our forward and backward trace operations, we're going to assume that a specific Windows computer has been compromised.

We want to build a chain of events from the compromised host to try to determine the identity of the hacker. To do this, we'll need to:

- 1. Identify the relevant login messages.
- 2. Give the compromised host as the first value to match.
- 3. Extract other relevant values (src_host, dest_host, login_user):

Tracing forward

We want to trace all Windows logins moving forward (+), starting from John's workstation (which may be compromised), to build a chain of events. We can use a trace operator query to produce the following results:

```
* "EventIdentifier = 4624" "\nLogon Type:\t\t\t10" OR "\nLogon Type:\t\t\t2" | trace + "(?:Computer|Workstation )Name(?: = \"|:|\\t)?(.*?)\" \"JohnWorkstation.example.com\" | extract \"ComputerName = \"(?:<dest_host>.+)\" | extract \"Workstation Name:\\t(?(<src_host>.+)\" | extract \"Name:\\t\\t\\t(?(<login_user>.*)\" | s"
```

Saved searches: [Open](#) | [Save](#) | [Save As](#)

06/08/2012 10:36:49 AM

8

6

4

2

10:40 AM10:50 AM

Status: Done gathering results Elapsed time: 00:00:01 Results: 3 Session: 050B5CE790FC7A5D

Messages

⏪

⏴

Page: 1 of 1

⏵

⏩

#	Time	dest_host	login_user	src_host
1	06/08/2012 10:37:57.126	WIN1.example.com	JOHN.DOE	JohnWorkstation.example.co
2	06/08/2012 10:37:57.126	WIN2.example.com	Administrator	WIN1.example.com
3	06/08/2012 10:38:01.906	WIN3.example.com	Administrator	WIN2.example.com

268

Tracing backward

*** "EventIdentifier = 4624" "\nLogon Type:\t\t\t10" OR "\nLogon Type:\t\t\t3" | trace - "(?:Computer\|Workstation)Name(?: = \"|;\t)?(.+?)(?:\" | extract "ComputerName = \"(?:<dest_host>.+)\" | extract "Workstation Name:\\t(?:<src_host>.+)\s" | extract "New Logon:[\s\S]+?Account Name:\\t**

Saved searches: [Open](#) | [Save](#) | [Save As](#)

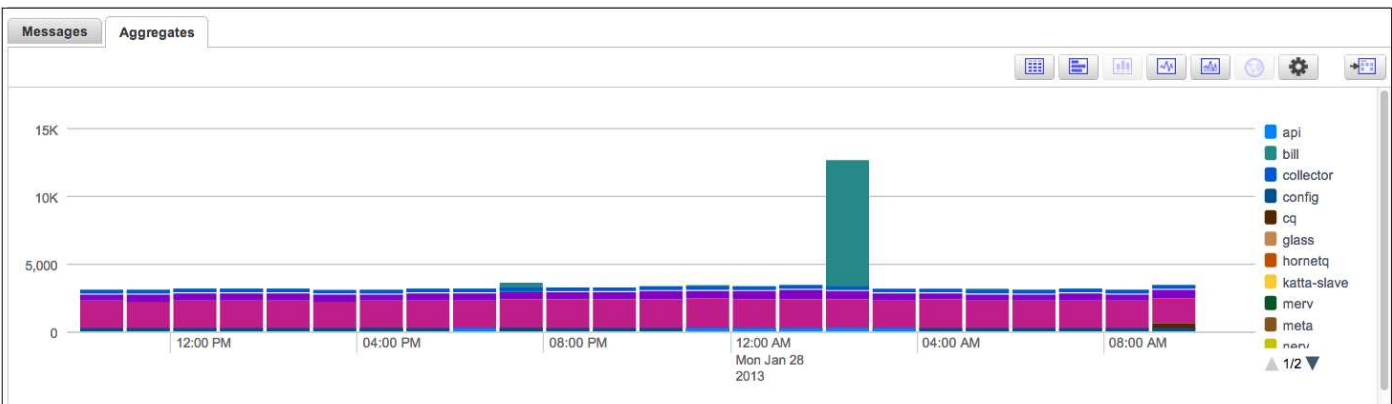
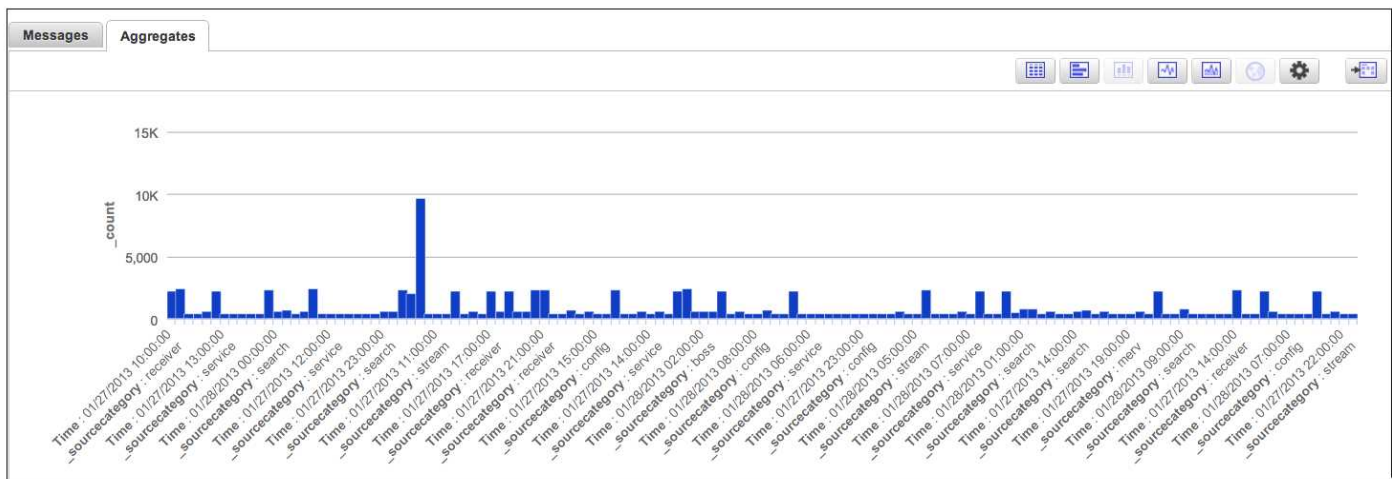
Status: Done gathering results Elapsed time: 00:00:02 Results: 3 Session: 5776126BB57C362A

Messages

#	Time	dest_host	login_user	src_host
1	06/08/2012 10:13:40.668	WIN3.example.com	Administrator	WIN2.example.com
2	06/08/2012 10:13:35.662	WIN2.example.com	Administrator	WIN1.example.com
3	06/08/2012 10:13:35.662	WIN1.example.com	JOHN.DOE	JohnWorkstation.example.com

Transpose Operator

Transpose operators are especially useful to include in Dashboard Monitor queries, because data can be displayed in a meaningful way. For example, the graphs below represent the same data from the same time range, but one is generated from a query using the transpose operator:



The bottom example, which is generated by a search that includes a transpose operator, results are displayed in a more understandable manner, meaning that the organization can use the data much more efficiently.

Syntax:

- transpose row [row fields] column [column fields] as [output fields]
- transpose row [row fields] column [column fields]

The above syntax is equivalent to transpose row [row fields] column [column fields] as *

Results can be influenced in three ways:

- By using a comma separated list of variable names (such as "a, b"), only the specified output fields appear in the output table.
- By using a comma separated list of variable names, followed by a comma and a star (such as "a, b,*"), the specified output fields appear in the output table, followed by dynamic fields.
- By including a single star ("*") all dynamic fields appear in the output.



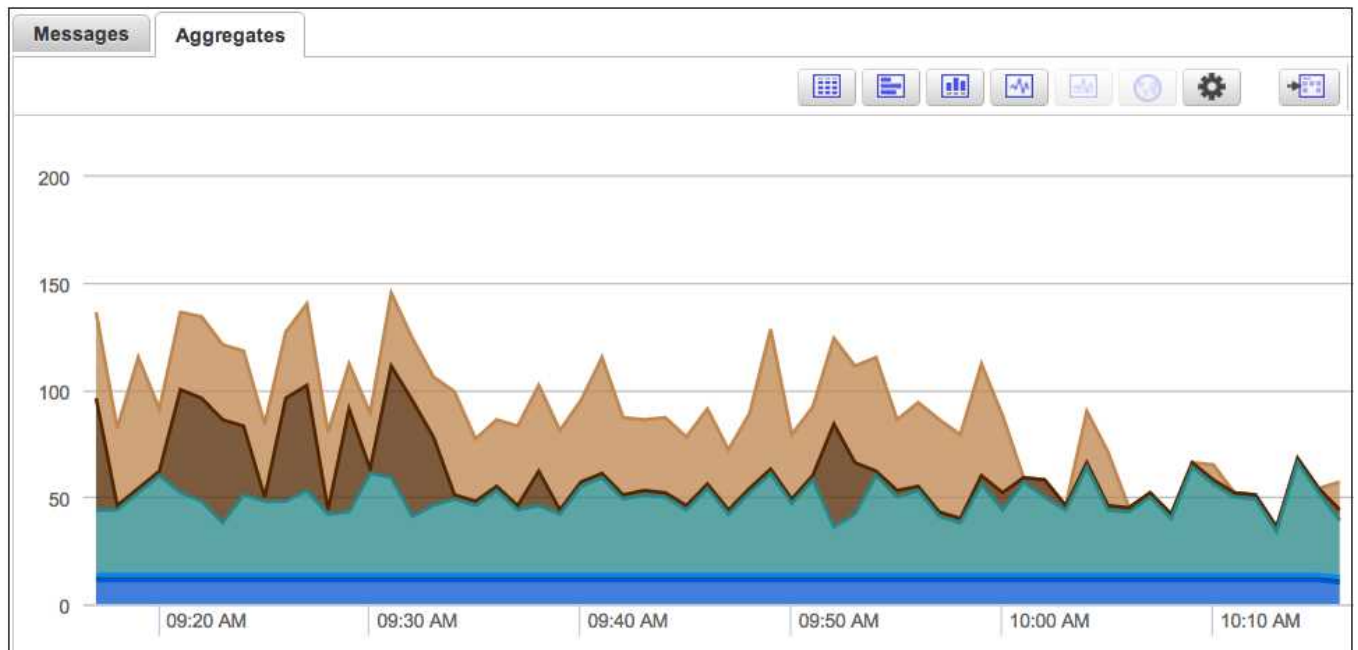
As a reminder, if a field name contains a special character (such as -) the character must be quoted in %"", as in %"test-zz-1". Because column names computed from data tend to include special characters, this is especially important to keep in mind when using a transpose operator.

Example:

Viewing errors by module. Let's say that errors are logged by module; we'd like to view errors by each module's name. Running a query similar to:

```
error | parse "module=*" as module
| timeslice 1m
| count as value by _timeslice, module
| transpose row _timeslice column module as [moduleName1, moduleName2, ...]
```

will produce results with each module represented with a distinct color, similar to:

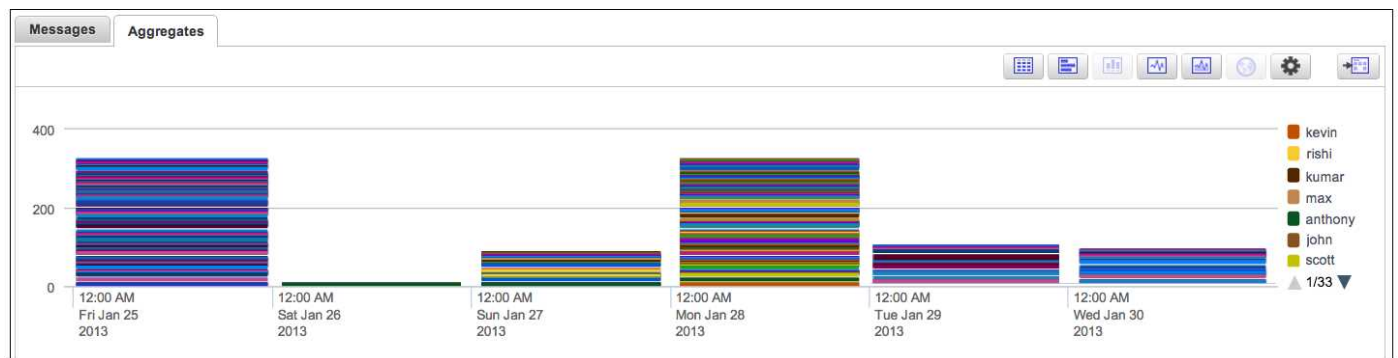


Try changing the Stacking setting (under Change Properties) to **Normal** to see how graphs are affected by this option. For more information, see [Graphing Search Results](#).

View successful logins by user. Because you can use the transpose operator without prior knowledge of the fields it will generate, you can view logins by users and organization. Running a query similar to:

```
_sourceCategory=service
| parse "Successful login for user **", organization: "*" as user, org_id
| timeslice 1d
| count _timeslice, user
| transpose row _timeslice column user
```

will produce a graph similar to:



For information on handling null fields, see [isNull operator](#).

URL Decode Operator

The URL Decode (`urldecode`) operator decodes a URL you include in a query, returning the decoded (unescaped) URL string.

For example, a URL that looks like this:

```
url%3D%2Fscience%2FUserActions%2Farticle%2FrightPane%2Fserial%2FdisplayRightPanePanel%2FciteRelatedArt...
```

can be decoded to:

```
url=/science/UserActions/article/rightPane/serial/displayRightPanePanel...
```

Syntax:

- `urldecode(url_to_be_decoded)`

Example:

Let's say you'd like to decode URLs connecting to your firewall. Running a query like:

```
http: | parse "Connecting to firewall at URL: *" as url | urldecode(url) as decoded
```

returns results of each URL, both in the encoded and decoded state, allowing you to run additional queries on the parsed, decoded URLs.

CHAPTER B

Sumo Logic Suggested Searches

To better understand how Sumo Logic queries can help track and diagnose common IT issues like monitoring traffic from devices, failed network logins, kernel failures, or many other issues, take a look at these Sumo Logic Suggested Searches:

- [Searches for the Apache Access Parser](#)
- [Searches for the Apache Errors Parser](#)
- [Searches for Cisco ASA Monitoring](#)
- [Searches for Linux OS Security, Audit, and Performance Issues](#)
- [Searches for the Microsoft IIS Parser](#)

Check back often to find additional Suggested Searches as we add them. For deeper information about Sumo Logic Search Syntax, see the [Search Syntax Reference](#).

Searches for the Apache Access Parser

The following searches were built for use with the Apache Access Parser. Copy and paste these searches into the search query field and save them for later use.

To obtain the best results possible, be sure to make the following modifications to the example queries:

- Use a specific keyword expression or metadata search to limit the initial results to Apache logs. [Replace _ the metadata search expression "sourceName=*error_log* AND _sourceCategory=*apache*" in the examples.]
- Change the time range and the timeslice values to tailor the results to your needs.

Understanding Incoming Requests

Use these searches for troubleshooting and performance monitoring.

All HTTP Response codes with their count

This search returns the number of requests observed per HTTP status code.

- Suggested Time Range: -1d
- `_sourceName=*access_log* AND _sourceCategory=*apache* | parse using public/apache/access | count by status_code | sort by _count`

Client Errors (4xx response codes) per day

Returns the number of errors caused by web clients over the past seven days. This search can be used to detect if these errors are increasing or decreasing over time. These kinds of errors can either indicate whether the number of attacks to the website are increasing (for example, 403 Forbidden responses) or whether website changes are causing more errors.

- Suggested Time Range: -7d
- `_sourceName=*access_log* AND _sourceCategory=*apache* | parse using public/apache/access | where status_code matches "4*" | timeslice by 1d | count by _timeslice`

Server Response Summary Over Time

Returns the number of client errors, server errors, redirects and successful responses observed each day over the last seven days. This search can be used to understand the distribution of errors vs. successful responses and redirects.

- Suggested Time Range: -7d
- `_sourceName=*access_log* AND _sourceCategory=*apache* | parse using public/apache/access | if (status_code matches "2*", 1, 0) as successes | if(status_code matches "5*", 1, 0) as server_errors | if (status_code matches "4*", 1, 0) as client_errors | timeslice by 1d | sum(successes) as successes, sum(client_errors) as client_errors, sum(server_errors) as server_errors by _timeslice`

Top 404 referrers

Returns the top 100 URLs that refer to a resource that does not exist on the website. This information can be used to fix existing web pages.

- Suggested Time Range: -1d
- `(_sourceName=*access_log* AND _sourceCategory=*apache*) AND "404" | parse using public/apache/access | where status_code="404" | count_frequent referrer | limit 100`

Top Clients Causing Errors

Returns the top source IP addresses that are responsible for client errors on the website, which can help in blocking certain IP addresses or IP address ranges from accessing the website.

- Suggested Time Range: -1d
- `_sourceName=*access_log* AND _sourceCategory=*apache* | parse using public/apache/access | where status_code matches "4*" | count_frequent src_ip | limit 100`

Understand Incoming Traffic

The following searches can help you get information about the volume of traffic and the amount of bytes served by your website over a period of time. In addition, the searches can help you learn more about the browsers and devices accessing your site.

Traffic volume and bytes served per day

Returns the number of bytes served and the number of hits to the website each day over the past day. If run over a longer period of time (say two weeks or a month), this also may give a good idea of which days of the week are more busy than others.

- Suggested Time Range: -7d
- `_sourceName=*access_log* AND _sourceCategory=*apache* | parse using public/apache/access | where size != "-" | (size/1048576) as mbytes | timeslice by 1d | sum(mbytes) as Bytes_Served_mb, count as hits by _timeslice`

Top clients

This search returns the top 100 IP addresses that cause the most hits to the website.

- Suggested Time Range: -1d up to -7d
- `_sourceName=*access_log* AND _sourceCategory=*apache* | parse using public/apache/access | count_frequent src_ip`

Top browsers

This search returns the top 100 browsers that are accessing the website.

- Suggested Time Range: -1d
- `_sourceName=*access_log* AND _sourceCategory=*apache* | parse using public/apache/access | count_frequent user_agent | limit 100`

Robots

This search returns a list of all robots that are accessing the website, assuming that robots first access the robots.txt file before actually crawling through the website.

- Suggested Time Range: -1d
- `(_sourceName=*access_log* AND _sourceCategory=*apache*) AND "/robots.txt" | parse using public/apache/access | count_frequent user_agent`

Understand Page-Served Time

Time taken to serve requests

This search adds the time taken in seconds, microseconds and minutes to serve the requests as additional fields. It can be used to get quick insight into the performance of the server at present.



This is assuming that the `%T/%D` logging directive is added to the access log where `%T` is the number of seconds and `%D` is the number of microseconds taken to serve the request.

- Suggested Time Range: -30m
- `_sourceName=*access_log* AND _sourceCategory=*apache* | parse using public/apache/access | extract "(?<seconds>\d+)/(?<microseconds>\d+)$" | toLong(microseconds/(1000000*60)) as minutes`

URLs and Bytes Served Statistics

Malicious URL requests

Get information on incoming client request URLs which are sent for malicious reasons. These may or may not cause client or server errors.

- Suggested Time Range: -2h
- `(_sourceName=*access_log* AND _sourceCategory=*apache*) AND ("jsessionid" OR "old" OR "bak") | parse using public/apache/access | where url matches "'.old'" OR url matches "'.bak'" OR url matches "**jsessionid=**"`

Top URLs by bytes served

- Suggested Time Range: -1d
- `_sourceName=*access_log* AND _sourceCategory=*apache* | parse using public/apache/access | where size != "-" | avg(size) as average_size_KB by url | sort by average_size_KB | limit 100 | (average_size_KB/1024) as average_size_KB | (toLong(average_size_KB*100)/100) as average_size_KB`

Searches for the Apache Errors Parser

The following searches were built for use with the Apache Errors Parser. Copy and paste these searches into the search query field and save them for use later. You can also set up [threshold alerts](#) for the Critical Operations Errors to be notified in the event critical errors occur.

To obtain the best results possible, be sure to make the following modifications to the example queries:

- Use a specific keyword expression or metadata search to limit the initial results to Apache logs. [Replace `_` the metadata search expression `"sourceName=*error_log* AND _sourceCategory=*apache*"` in the examples]
- Change the time range and the timeslice values to tailor the results to your needs.

Identify Critical Operations Errors

Critical Log Messages

This search returns the most critical log messages in the Apache error log.

- Suggested Time Range: -10m; Set to run every 15 minutes with an alert for 1 or more results found.
- `_sourceName=*error_log* AND _sourceCategory=*apache* AND ("emerg" OR "alert" OR "crit") | parse using public/apache/error`

Log Level Counts

Returns a count of all messages by log level (error, warn etc.) to give administrators quick insight into whether they need to investigate further.

- Suggested Time Range: -15m
- `_sourceName=*error_log* AND _sourceCategory=*apache* | parse using public/apache/error | where log_level != "" | count by log_level | sort by _count`

Server Stops and Starts Over Time

Returns trend data of how many server start and stop events took place over a period of time.

- Suggested Time Range: -6h
- `_sourceName=*error_log* AND _sourceCategory=*apache* | parse using public/apache/error | if(reason matches "caught SIGTERM, shutting down", 1, 0) as server_stop | if(reason matches "-- resuming normal operations", 1, 0) as server_start | timeslice by 1h | sum(server_stop) as server_stops, sum(server_start) as server_starts by _timeslice`

Identify Top Error Characteristics

Top Error Reasons

This search returns the top Apache error log reasons.

- Suggested Time Range: -6h
- `_sourceName=*error_log* AND _sourceCategory=*apache* | parse using public/apache/error | count by reason | top 100 reason by _count`

Top Clients Causing Errors

Returns the top source IP addresses that cause errors, which should correlate with the corresponding [access log searches](#) to determine the most malicious clients.

- Suggested Time Range: -6h
- `_sourceName=*error_log* AND _sourceCategory=*apache* | parse using public/apache/error | count_frequent src_ip | limit 100`

Searches for the Cisco ASA Parser

These suggested searches cover some of the most common scenarios for monitoring Security, Audit, and Performance issues on a Linux server. You can enter these queries into the Search field of the Sumo Logic Web Application as a starting baseline, and then customize the queries for your system. Be sure to [save your search queries](#) if you plan to run them often.

These are a few valuable search queries you can enter in the Search field when you want to discover details about your Cisco ASA traffic.



The `_sourceCategory` fields shown in these sample queries are based on [Sumo Logic's recommendations for adding metadata to Sources](#). To re-use these queries, type the Category you entered for the relevant Source after "`_sourceCategory=`" or use an asterisk wildcard (*) instead.

Top Denied Sources

Returns the top sources that were denied.

- Suggested Time Range: -1h
- `_sourceCategory=*cisco*asa* AND ("denied" OR "Deny") | parse using public/cisco/asa | where access_decision="denied" OR action matches "Deny *" | count_frequent src_host | limit 10`

Top Denied Destinations

Returns the top destinations that were denied.

- Suggested Time Range: -1d
- `_sourceCategory=*cisco*asa* AND ("denied" OR "Deny") | parse using public/cisco/asa | where access_decision="denied" OR action matches "Deny *" | count_frequent dest_host | limit 10`

Top Sources with Outbound Connections

Returns the top sources with outbound connections by the number of connections.

- Suggested Time Range: -1h
- `_sourceCategory=*cisco*asa* AND "built outbound" | parse using public/cisco/asa | where src_host != "" | count_frequent src_host | limit 10`

Top Internal Destinations

Returns the top internal destinations by number of connections.

- Suggested Time Range: -1h
- `_sourceCategory=*cisco*asa* AND "built inbound" | parse using public/cisco/asa | where dest_host != "" | dest_host as internal_destination | count_frequent internal_destination | limit 10`

Detected Attacks

Returns all attacks detected by the IPS.

- Suggested Time Range: -15m; run as a scheduled search to return results only if number of messages is > 0
- `_sourceCategory=*cisco*asa* ": ips:" AND ("attack" OR "Proxied RPC Request" OR "buffer overflow" OR "IP Impossible Packet" OR "IP Fragments Overlap" OR "Fragmented ICMP Traffic" OR "Large ICMP Traffic" OR "TCP NULL flags" OR "TCP SYN+FIN flags" OR "TCP FIN only flags") | parse using public/cisco/asa`

Suggested Searches for Linux OS Systems

These suggested searches cover some of the most common scenarios for monitoring user activity and security activity on a Linux server. These searches should work on RedHat, Debian, SuSe platforms, as well as their derivations (for example, CentOS, Ubuntu, OpenSuSe).

You can enter these queries into the Search field of the Sumo Logic Web Application as a starting baseline, and then customize a query and time range for your system. Be sure to save your search queries if you plan to run them often.

It's assumed that common Linux OS logs are collected (for example: `/var/log/*`).

The `_sourceCategory` fields shown in these sample queries are based on the following Linux logs and their metadata:

- **Generic system log:** Typically named `/var/log/syslog` or `/var/log/messages`
Meta field: `SourceCategory = OS/Linux/System`
- **Authentication log:** Typically named `/var/log/auth` or `/var/log/auth.log`
Meta field: `SourceCategory=OS/Linux/Security`

These logs might have also been collected by the Collector (if selected during its installation).

User activity

These searches are intended to help you understand how privileged and non-privileged users are authenticating to and using your Linux servers.

Successful User Login events

Returns all successful remote and local logins by a user.

- Suggested time range: -1 day
- `_sourceCategory=OS/Linux/Security ("su:" or "sudo:" or "sshd:" or "sshd[" or "pam:") (("Accepted" and "pam") or "session" or ("to" and "on")) !"closed"`
`| parse regex "\S*\s+\d+\s+\d+:\d+:\d+\s(?:<dest_hostname>\S*)\s(?:<process_name>\w*):\s+(?:<message>.*)" nodrop`
`| parse regex "\S*\s+\d+\s+\d+:\d+:\d+\s(?:<dest_hostname>\S*)\s(?:<process_name>\S*)\s(?:<message>.*)" nodrop`
`| parse "session * for user * by *(uid=*)" as (action,dest_user,src_user,src_user_id) nodrop`
`| parse regex "session (?:<action>\w*) for user (?:<dest_user>\S*)" nodrop`
`| parse "Accepted keyboard-interactive/pam for * from * port * *" as (dest_user,src_hostname,src_port,protocol)`
`| where dest_user!=""`

All failed authentication attempts

Returns all failed authentication attempts by either a user or a process.

- Suggested time range: -1 day
- `_sourceCategory=OS/Linux/* "Authentication failure"`
`| parse regex "\S*\s+\d+\s+\d+:\d+:\d+\s(?:<dest_hostname>\S*)\s(?:<process_name>\w*)(?:\s(?:<message>.*))?" nodrop`
`| parse " user = * " as dest_user nodrop`
`| parse "User *: Authentication failure" as dest_user nodrop`
`| parse " user=" as dest_user nodrop`

Root activities

Returns all sudo/su attempts, or activities by "root" user. Modify to include other privileged users that you want to track in your environment.

- `_sourceCategory=OS/Linux/Security ("sudo" or "root" or "su")`
`| parse regex "\S*\s+\d+\s+\d+:\d+:\d+\s(?:<dest_hostname>\S*)\s" nodrop`
`| extract "sudo:\s+(?:<src_user>[^\s]+\s):.+?USER=(?:<dest_user>[^\s]+\s)" nodrop`
`| parse regex "COMMAND=(?:<command>[^\s]*)$" nodrop`
`| parse " user * " as dest_user nodrop | parse " user=" as dest_user nodrop`
`| where command != "" or dest_user in ("root") or src_user in ("root")`

Failed SU attempts

Returns all failed SU attempts.

- `_sourceCategory=OS/Linux/Security ("authentication failure" or "FAILED SU") ("su:" or "su["`
`| parse regex "\S*\s+\d+\s+\d+:\d+:\d+\s(?:<dest_hostname>\S*)\s" nodrop`
`| parse "ruser=* rhost=* user=" as src_user,src_hostname, dest_user nodrop`
`| parse "Authentication failure for * from *" as dest_user,src_hostname nodrop`
`| parse "FAILED SU (to *) * on" as dest_user,src_user nodrop`
`| parse "FAILED su for * by *" as dest_user,src_user nodrop`
`| where dest_user!=" " and src_user!=" "`

Security activity monitoring

New users

Returns a list of all new users.

- Suggested time range: -1 day
- `_sourceCategory=OS/Linux/S* "useradd" and (("new user") or ("new account"))`
 - | parse regex "`\S*\s+\d+\s+\d+:\d+:\d+\s(?:<dest_hostname>\S*)\s(?:<process_name>\w*)(?:\[:])`" nodrop
 - | parse "name=*, UID=*, GID=*, home=*, shell=*" as dest_user,dest_uid,dest_gid,home_dir,shell nodrop
 - | parse "account=*, uid=*, gid=*, home=*, shell=*" as dest_user,dest_uid,dest_gid,home_dir,shell nodrop

New groups

Returns a list of all new groups.

- Suggested time range: -1 day
- `_sourceCategory=OS/Linux/S* "new group"`
 - | parse regex "`\S*\s+\d+\s+\d+:\d+:\d+\s(?:<dest_hostname>\S*)\s(?:<process_name>\w*)(?:\[:])`" nodrop
 - | parse "name=*, GID=*" as dest_group,dest_gid nodrop
 - | parse "group=*, gid=*" as dest_group,dest_gid nodrop

Existing users added to privileged groups

Returns all messages that indicate a user being added to an administrative group. **Modify this query to include the IDs or names of the administrative groups in your environment.**

- Suggested time range: -1 day
- `_sourceCategory=OS/Linux/S* "to group" or "default group changed" or "change user"`
 - | parse regex "`\S*\s+\d+\s+\d+:\d+:\d+\s(?:<dest_hostname>\S*)\s(?:<process_name>\w*)(?:\[:])`" nodrop
 - | parse "add '*' to group '*'" as dest_user,dest_group nodrop
 - | parse "account added to group - account=*, group=*, gid=*" as dest_user,dest_group,dest_gid nodrop
 - | parse "account=*, uid=*, gid=*, old gid=*" as dest_user,dest_uid, dest_gid,src_gid nodrop
 - | parse "change user '*' GID from '*' to '*'" as dest_user,src_gid, dest_gid nodrop
 - | where dest_gid in ("10","0","4") or dest_group in ("root", "wheel", "adm")

Failed Password Changes

Returns all failed attempts to change a user password.

- Suggested time range: -1 day
- `_sourceCategory=OS/Linux/* "Authentication failure"`
 - | parse regex "`\S*\s+\d+\s+\d+:\d+:\d+\s(?:<dest_hostname>\S*)\s(?:<process_name>\w*)(?:\[:])`" nodrop
 - | parse "User *:" as dest_user nodrop
 - | parse " user=*" as dest_user nodrop
 - | where process_name="passwd"

System Start

Returns all incidents when the sytem starts (or restarts).

- Suggested time range: -1 day
- `_sourceCategory=OS/Linux/System "Initializing cgroup subsys cpuset"`
| parse regex `"^(?<StartTime>S*\s+\d+\s+\d+:\d+:\d+)\s(?<dest_hostname>S*)\s(?<process_name>w*)(?:\[\d+\])?:\s+" nodrop`

Service Shutdown/Exiting

Returns all instances when a service is shutting down or exiting. (Note that this query cannot capture the cases when there is no log when a service is down.)

- Suggested time range: -1 day
- `_sourceCategory=OS/Linux/System ("exiting" or "exited" or "terminating" or "terminated" or "shutting")`
| parse regex `"S*\s+\d+\s+\d+:\d+:\d+\s(?<dest_hostname>S*)\s(?<process_name>w*)(?:\[\d+\])?:\s+"`
| where process_name != ""

Searches for the Microsoft IIS Parser

The following searches were built for use with the Microsoft IIS Parser. Copy and paste these searches into the search query field and save them for use later.

These are written assuming the messages are parsed by our out of the box IIS W3C access log parser (i.e. parse using public/iis).

HTTP Status Code Summary Over Time

Returns the number of client errors, server errors, redirects, and successful responses observed each day over the last seven days. This search can be used to understand the distribution of errors vs successful responses and redirects.

- Suggested Time Range: -7d
- `_sourceCategory=*IIS* | parse using public/iis | if(sc_status matches "2*", 1, 0) as successes | if(sc_status matches "3*", 1, 0) as redirects | if(sc_status matches "5*", 1, 0) as server_errors | if(sc_status matches "4*", 1, 0) as client_errors | timeslice by 1d | sum(successes) as successes, sum(redirects) as redirects, sum(client_errors) as client_errors, sum(server_errors) as server_errors by _timeslice`

Top 404 URLs

Returns the top 100 URLs that refer to a resource (that doesn't exist on the website). This information can be used to fix existing web pages.

- Suggested Time Range: -1d
- `_sourceCategory=*IIS* "404" | parse using public/iis | where sc_status matches "404" | count_frequent cs_uri_stem | limit 100`

Traffic Volume Served Per Day

Returns the number of hits on a website each day over the past 24 hours. If this search is run over a longer period of time (such as two weeks or a month) it may give you a good idea of which days of the week are busier.

- Suggested Time Range: -7d
- `_sourceCategory=*IIS* | parse using public/iis | timeslice by 1d | count as hits by _timeslice`

Top Browsers

Returns the top 10 browsers accessing the website.

- Suggested Time Range: -1d
- `_sourceCategory=*IIS* | parse using public/iis | count_frequent cs_user_agent | limit 10f`

Slowest URLs by Average Time

- Suggested Time Range: -1d
- `_sourceCategory=*IIS* | parse using public/iis | (time_taken/1000) as seconds | avg(seconds) as avgttimeinseconds by cs_uri_stem | sort by avgttimeinseconds | limit 100`

Searches for Windows 2008 Events

Domain Controller/Windows Server Events

We recommend saving the following searches and scheduling them to run every 10 minutes over the last 10 minutes.

Detect when the audit policy was cleared

- Suggested Time Range: -10m
- `_sourceCategory=OS/Windows 1102 | parse using public/windows/2008 | where event_id="1102"`

Detect when the audit policy was changed

- Suggested Time Range: -10m
- `_sourceCategory=OS/Windows 4719 | parse using public/windows/2008 | where event_id="4719"`

Detect account policy changes

This event indicates that the computer's Security Settings\Account Policy or Account policy was modified, either via Local Security Policy or Group Policy in Active Directory. Similar to the audit policy, the Account Policy should not change often, making this a critical event to monitor.

- Suggested Time Range: -10m
- `_sourceCategory=OS/Windows 4739 | parse using public/windows/2008 | where event_id="4739"`

Detect system restarts

This event indicates a machine restart, and is important to monitor because production systems should never go down.

- Suggested Time Range: -10m
- `_sourceCategory=OS/Windows 4608 | parse using public/windows/2008 | where event_id="4608"`

Detect service installation

This events indicates that a new service was installed in the system. Any new service installed on the domain controller needs to be monitored closely.

- Suggested Time Range: -10m
- `_sourceCategory=OS/Windows (4946 OR 4947 OR 4948) | parse using public/windows/2008 | where event_id="4946","4947","4948"`

Logon/Logoff Events

Failed logins on the Domain Controller

These events indicate failed logins, either on the Domain Controller or on member servers.

- Suggested Time Range: -1h
- `_sourceCategory=OS/Windows (4771 OR 4768 OR 4776) | parse using public/windows/2008 | where event_id="4771" OR (event_id="4768" AND result_code != "0x0") OR event_id="4776"`

Consecutive failed logins by the same user

These events indicate consecutive failed logins whether on the Domain Controller or on member servers.

- Suggested Time Range: Save as a search; set the search to run every 15 minutes only if the number of groups is greater than 0.
- `_sourceCategory=OS/Windows (4771 OR 4768 OR 4776) | parse using public/windows/2008 | where event_id="4771" OR (event_id="4768" AND result_code != "0x0") OR event_id="4776"`

User account changed

This event is logged when a user account is changed.

- Suggested Time Range: -1d
- `_sourceCategory=OS/Windows 4738 | parse using public/windows/2008 | where event_id="4738"`

User added to a group

These events are logged when a user is added to a group.

- Suggested Time Range: -1d
- `_sourceCategory=OS/Windows (4728 OR 4732 OR 4756) | parse using public/windows/2008 | where event_id in ("4728", "4732", "4756")`

New accounts created

This event is logged when new accounts are created.

- Suggested Time Range: -1d
- `_sourceCategory=OS/Windows 4720 | parse using public/windows/2008 | where event_id="4720"`

User locked out

This event is logged when a user is locked out after repeated logon failures.

- Suggested Time Range: -15m; Save as a search to run every 15 minutes if the number of messages is less than one.
- `_sourceCategory=OS/Windows 4740 | parse using public/windows/2008 | where event_id="4740"`

Successful logins over time

This search returns the total number of successful logins on a local machine every hour, over the past 24 hours.

- Suggested Time Range: -24h
- `_sourceCategory=OS/Windows 4624 | parse using public/windows/2008 | where event_id="4624" | timeslice by 1d | count by _timeslice`

Consecutive failed logins on a local machine

- Suggested Time Range: -15m; Save as a search to run every 15 minutes and return results if the number of messages is >1.
- `_sourceCategory=OS/Windows 4625 | parse using public/windows/2008 | where event_id="4625" | count by target_acct | where _count > 3`

Top reasons for failed logins

This search returns the top reasons why logins are failing.

- Suggested Time Range: -6h
- `_sourceCategory=OS/Windows 4625 | parse using public/windows/2008 | where event_id="4625" | count by dest_user | top 10 fail_reason by _count`

User/Account Changes

User password reset attempt

This search returns all instances where an attempt has been made to reset a user's password. This event is triggered by someone else changing a user's password (not the user himself).

- Suggested Time Range: -6h
- `_sourceCategory=OS/Windows 4724 | parse using public/windows/2008 | where event_id="4724"`

User password changes

This search returns the number of times user have changed their own passwords each day over the past week.

- Suggested Time Range: -7d
- `_sourceCategory=OS/Windows 4723 | parse using public/windows/2008 | where event_id="4723" | timeslice by 1d | count by _timeslice`

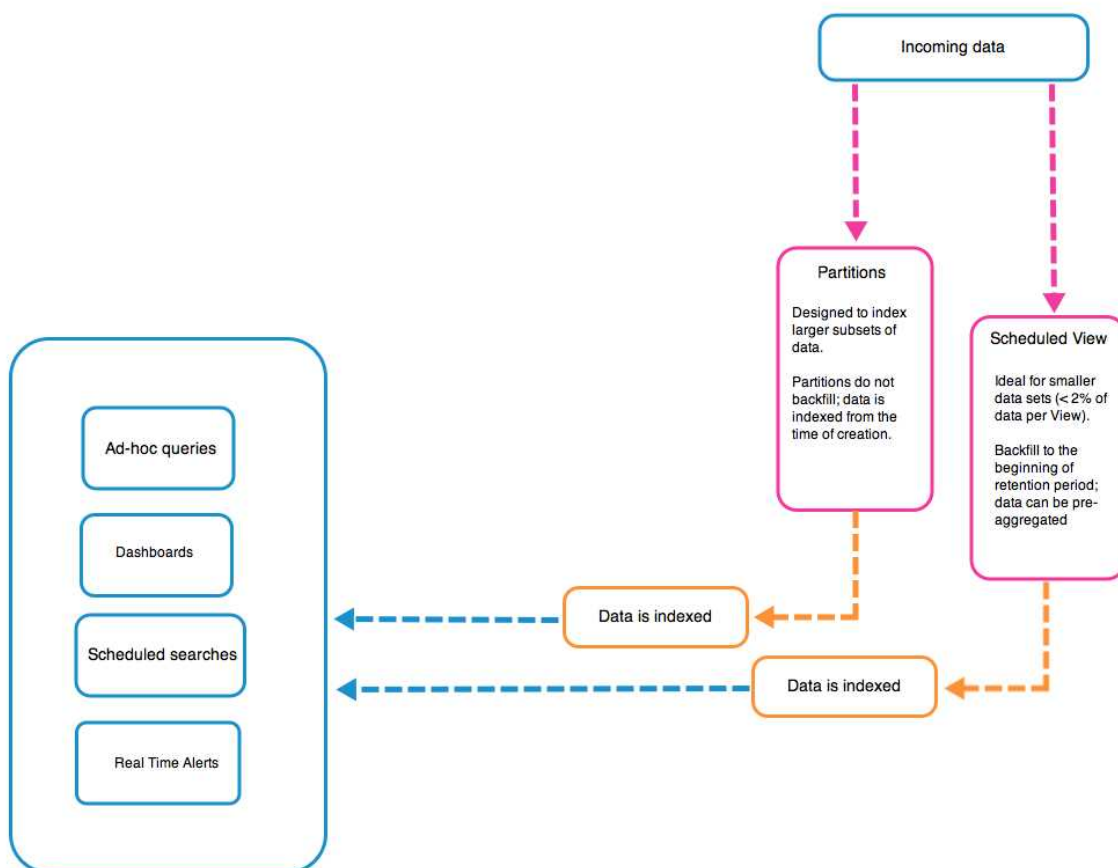
Search Optimization

Seeing quick search results is important to all Sumo Logic users. Generally speaking, the speed at which results displayed depended on the amount of data and the type of query run against that data. With search optimization tools, you can see significant improvement on search speeds.

The key to search optimization is to design a method of segmenting data in such a way that it's queued up for quick results. Sumo Logic has two methods of search optimization: **Partitions** and **Scheduled Views**.

Both of these search optimization tools are build around the concept of an index. An **index** is a subset of data. When you run a search against any type of index, because it's a smaller silo of data, search results are returned more quickly and efficiently.

Additionally, Sumo Logic provides a Data Volume index and App that are pre-configured to display information directly related to the amount of data your Sumo Logic account is processing. Although this is technically an index, it isn't customizable. You can learn more in [Data Volume Index](#).



Does all data need to be indexed?

No, not all data needs to be routed to an index. Any data not assigned to an index is saved to what's referred to as the default index. Think of the default index as the same Sumo Logic you've always used, where raw data is searched when a query is run.

Is there such a thing as creating too many indexes?

Yes. Indexes can be overused, and in some situations this can slow search results. When designing your organization's indexes, make sure to put some thought into the minimal amount of data that makes sense to index, no matter which tools you're using. When running a search on non-indexed data, Sumo Logic may need to process all indexed data as well, which can take quite a while.

Choosing the right search optimization tool

The use cases for Partitions and Scheduled Views are quite different. Partitions are able to index much larger quantities of data; Scheduled Views should only be used for smaller indexes. Here's a quick look at when you should choose one tool or the other.

What I want to do is...	Partition	Scheduled View
Run queries against a certain set of data.	Pick me if the amount of data is more than 2%	Pick me if the amount of data you'd like to segregate is 2% or less.
Use data to identify long-term trends.		Pick me!
Segregate data by sourceCategory	Pick me!	
Have aggregate data ready to query.		Pick me!
Use RBAC to deny or grant access to the data set.	Pick me!	Pick me!

Partitions

Sumo Logic allows you to filter a subset of the messages in an Index into a Partition on ingest. Partitions are used for both security and performance use cases. For security, you could create a Partition for messages pertaining to, for example, a certain group within your organization, like HR, and limit access to that Partition based on the Role-Based Access Controls (RBAC) for your users.

Partitioning messages in an Index also improves search query performance, since by creating a Partition, you have reduced the total number of messages that need to be searched. Once messages are routed to a Partition, you can limit your search to those messages using the Partition name in a search query, such as `_index=index_name`. But all Partition Indices are automatically included in searches.

Partitions ingest your messages in real time using Continuous Query technology. This means that a Partition only ingests messages from the time it is created and going forward. Data is not back filled into a Partition. If a message is ingested into multiple Partitions, it will be duplicated.

In Sumo Logic, you create Partitions using simple search expressions, also called routing expressions. Once created, the Partition routing expression filter condition is applied to all incoming messages as they enter the system, and if the filter matches the message, it is put in the Partition. This dialog is accessible via the **Manage > Partitions** page. Once you create a Partition, it becomes available to be used with searches, Monitors and Dashboards, and RBAC.

For more information, see [Managing Partitions](#).

How are Partitions different from Sumo Logic Indices and Scheduled Views?

Partitions are different from Scheduled Views in that they do not back fill with aggregate data. Partitions begin building a non-aggregate index from the date a Partition is created, only indexing data moving forward.

Scheduled Views back fill with aggregate data, meaning that all data that extends back to the start date of the View query is added to the View.

Sumo Logic Indices are automatically created by Sumo Logic to deliver a specific data set that cannot be edited. Sumo Logic Indices must be manually enabled by an admin. Once enabled, volume data is NOT back filled to any time before the feature was enabled. Data is only provided from the time the feature is enabled forward.

Running a Search Against a Partition

Running a search against the data in a Partition is almost exactly the same as running any other query. The difference you'll notice is the speed at which results are returned, especially if you're searching over a large amount of data.

Queries that contain Partitions can be saved as scheduled searches, as Dashboard Monitors, and as published or saved searches.

To run a search against a Partition:

1. Go to **Manage > Partition**, then from the **Partition** page, copy the name of the Partition.
2. Click **Search** in the top ribbon to open the **Search** page.
3. In the search text box, type `_index=`, then paste or type the name of the Partition, like: `_index=[index-name]`.
4. Complete the query, or run the search just on the Partition.

Running search across multiple Partitions

You can run a query against multiple Partitions simultaneously in a few ways.

Using wildcards to search across multiple Partitions

The name of a Partition is basically a metadata tag. That means that you can use wildcards just as you would in any regular query. Let's say you have Partitions named `web_logs`, `apache_error`, `apache_access`, `pages`, and `tomcat`. With those Partition names in mind, you could do any of the following:

- Run a search like `_index=*` to return results from all of the Partitions.
- Enter `_index=apache*` or even `_index=a*` to run a search against Apache logs.
- Enter `_index=*a*` to return results from `apache_error`, `apache_access`, `pages`, and `tomcat`.

Using Booleans to search across multiple Partitions

Booleans in queries that use Partitions work the same way as they work in all other searches. For example, you'd enter `_index=apache_error OR _index=apache_access` to return results from just those two Partitions.

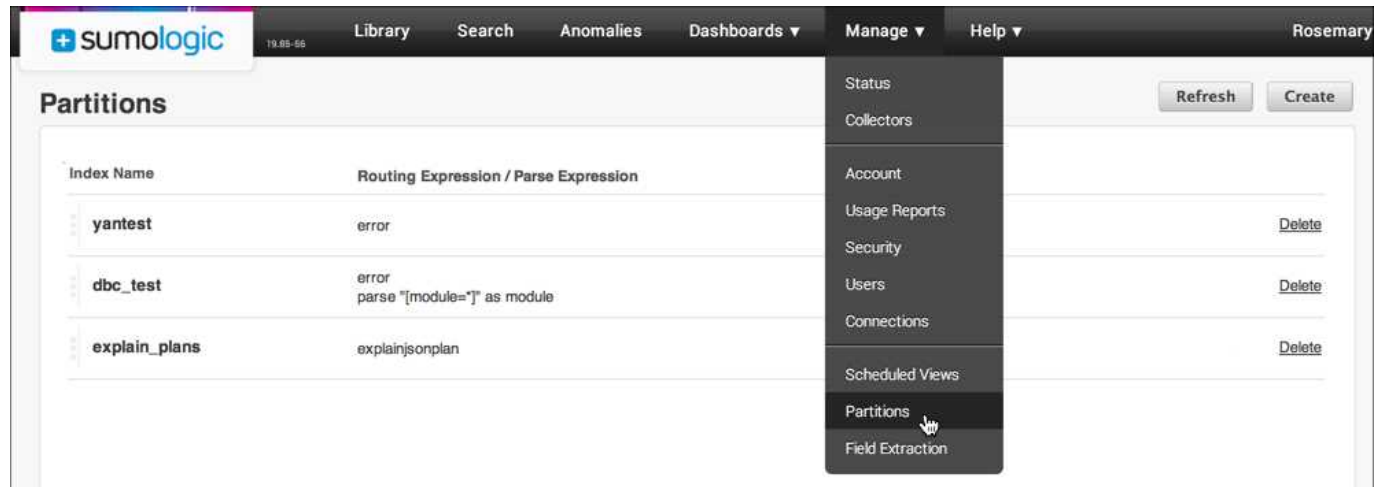
Using Partitions with Monitors and Dashboards

Once you have created a Partition, you can use it to increase search performance for Monitors and Dashboards that may benefit from using the Partition.

To use a Partition with a Monitor or Dashboard, just edit the query of the Monitor or Dashboard, by adding `_index=[index-name]`.

For details on editing the query, see [Saving edits to a Monitor](#).

Managing Partitions



The **Partitions** page displays the existing Partitions in a list, including the Partition name and the routing expression. On this page you can [create](#), [refresh](#), and delete, or [apply RBAC](#) to Partitions.

For conceptual information, see [Partitions](#).

Creating a Partition

To create a Partition in an Index, you will create a routing expression, which is a kind of query. A Partition routing expression can take anything in a regular search query up to the first pipe—in other words, the search constraints. Partitions must be named alphanumerically, with no special characters. The query can include wildcards, but it cannot include any parsing or search operators.

Create Partitions for use cases that are not too general. The idea is to use Partitions in an Index to restrict your search for security and in order to improve search performance. If you create a Partition for a very general use case, it would still work, you just wouldn't benefit as much from increased performance.

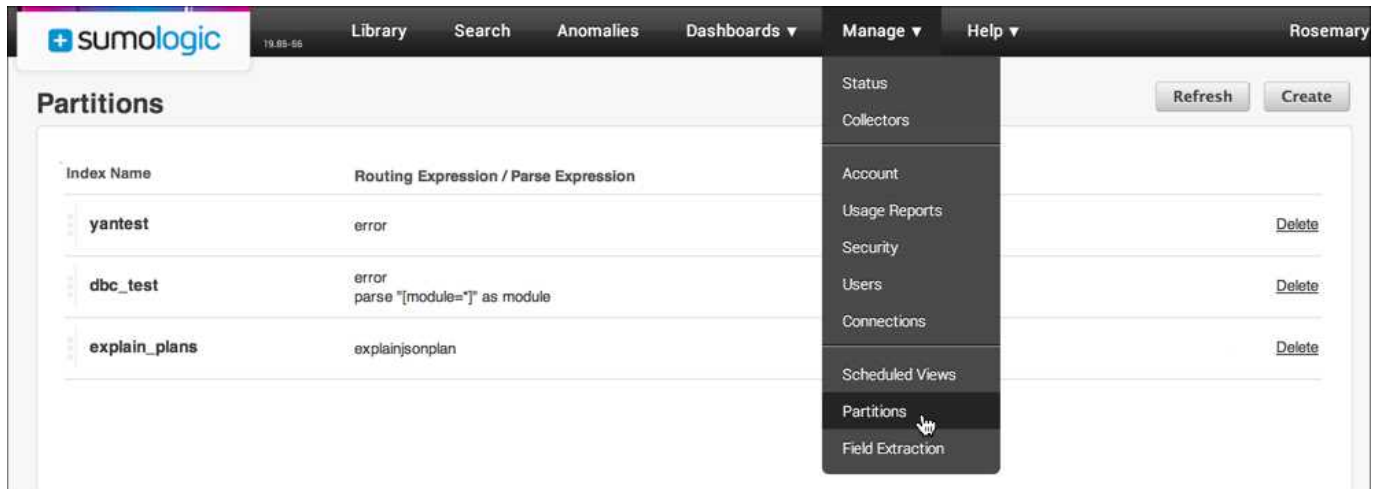
For example, you could create a Partition for all messages in an index that would pertain to the the Human Resources group in your organization, which might require additional security. Then you would use Role-Based Access Controls (RBAC) to restrict that Partition to users set with the HR role in your system. Or you could create a Partition for a certain type of log message, such as Apache logs.

When designing Partitions, use the following best practices:

- **Avoid using queries that are subject to change.** In order to benefit from using Partitions, they should be used for long-term message organization.
- **Make the query as specific as possible.** Making the query specific reduces the amount of data in the Partition, which increases search performance.
- **Keep the query flexible.** Use a flexible query, such as `sourceCategory=*Apache*`, so that meta data can be adjusted without breaking the query.
- **Group data together that is most often used together.** For example, create Partitions for categories such as web data, security data, or errors.

- **Group data together that is used by teams.** Partitions are an excellent way to organize messages by role and teams within your organization.
- **Avoid including too much data in your Partition.** For example, including 90% of the data in your index in a Partition won't improve search performance.

Adding a Partition



To create a Partition:

1. In the Sumo Logic Web Application, choose **Manage > Partitions**.
2. Click **Create**.

3. In the **Create a Partition** dialog box, enter the following:
 - **Index Name.** Enter a name that you'll use to search the data in a query. It's important to use a name that is descriptive and easy to remember. Names can be comprised of alphanumeric characters; underscores (_) are the only special characters allowed.
 - **Routing Expression.** Enter the routing query for the Partition, which consists of a keyword search and a filter condition. The routing query can include wildcards, but it cannot use any parsing or search operators. Also, empty strings are not supported.
4. Click **Create**.

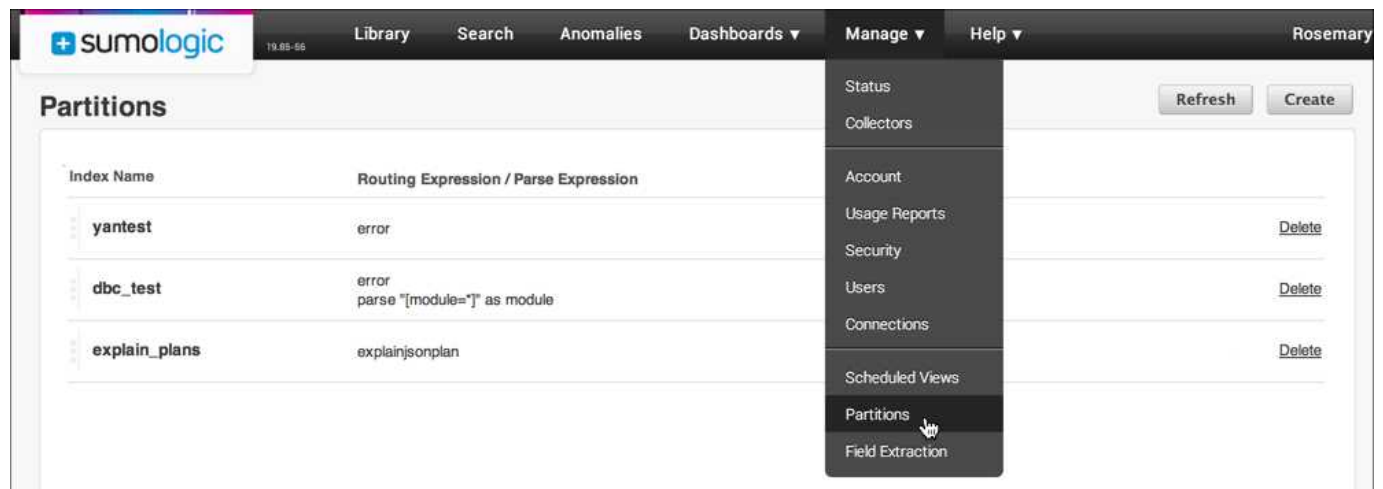
The new Partition is added to the list and begins to index data as soon as you create it. Allow a few hours for the indexing to complete. If you've chosen to index a large amount of data, it could take a bit longer.

How do new Partitions affect your current system?

Once you have created a new Partition, you may want to review the searches or Dashboards in your Sumo Logic environment to see how you might benefit from using the Partition in those queries, especially if you are querying only data in that Partition.

To do so, add `_index=[index-name]` to the query for the search or Dashboard.

Refresh or Delete Partitions



The **Partitions** page displays the existing Partitions in a list, including the Partition name and the routing expression. It also includes the following buttons and actions:

- **Refresh.** Refreshes the list of Partitions from the server.
- **Delete.** Deletes a selected Partition. Any admin can delete any Partition

Once a Partition has been created, it cannot be edited.

About Scheduled Views

A Scheduled View is a pre-aggregated index of a subset of data. After building a Scheduled View, you'll be able to run queries against that data set. Because the data is pre-aggregated, meaning that query you'll use to create a Scheduled View contains an aggregate function, search results return much quicker. Additionally, queries run against a Scheduled View cannot time out. Queries that run against Views can be used in scheduled searches, Dashboards, and in ad hoc searches.

The ability to run a query against historical data in a View means your team can uncover long-term trends and build Dashboards that include a large amount data without sacrificing performance. You can include data dated to the very beginning of your retention period. For example, if your organization has a 60-day retention period, you can use data from two months ago in your searches.

Because Scheduled Views add data on a one minute rolling schedule, you'll know that search results include recent log messages. Think of a Scheduled View as query that uses a one-minute timeslice to aggregate data. If you run a 60-minute search against a Scheduled View, you can expect 60 results (one for each one-minute aggregation).

How data is added to a Scheduled View

As data is being ingested into Sumo Logic, it's constantly being checked for how it should be handled. First, data is routed to any Partitions where it should be indexed. Then, data is checked against Scheduled Views; any data that matches the Views are indexed.

Data can be in a Partition and in a Scheduled View because the two tools are used differently (and are indexed separately). And, even though Partitions are indexed first, this architecture does not slow the indexing of Scheduled Views. Every minute, the query is run against the data routed to the Scheduled View, and then the results are indexed.

How are Scheduled Views different than Partitions and Sumo Logic Indices?

Scheduled Views are different from Partitions in that they **backfill** with **aggregate** data, meaning that all data that extends back to the start date of the View query is added to the View.

Partitions, however, begin building a **non-aggregate** index from the date a Partition is started, only indexing data moving forward. Sumo Logic Indices are automatically created by Sumo Logic to deliver a specific data set that cannot be edited.

Designing Scheduled Views

Scheduled Views are great for identifying long term trends. With that in mind, it's important to consider the uses that make the most sense for your organization, and build out a set of Scheduled Views that are general enough to be practical, yet specific enough to provide targeted search results.

How could your organization use Scheduled Views?

Web access trends. Creating a Scheduled View allows you to isolate logs related to your site, making it easy to report on web traffic patterns.

App usage metrics. A View can help you track the usage of one or more applications over time. Depending on your deployment, you could build a View per application.

Threat analysis. Because a Scheduled View indexes any type of data, you can create a View just for PAN logs, for example, that you can leverage to see how threat types and threat levels vary over time, or even which IPs from high-risk areas are hitting your site.

User behavior. Creating a View to parse out logins by user ID across your entire deployment means that you can answer audit-related questions quickly. Additionally, you can investigate behavior from a high-level with much faster results, like being able to see if users have logged in during the past 60 days (or as far back as your retention period).

Creating a Scheduled View

Admins can create Scheduled Views in the Sumo Logic Web Application. A Scheduled View is defined using a query, just as you'd use to run a regular search in Sumo Logic.

Keep the following in mind when you're constructing the query for the Scheduled View:

- **Avoid using queries that are likely to change.** A key benefit of using Scheduled Views is that they can index historical data, allowing you to identify long-term trends. If a query changes, you may lose some of the historical perspective.
- **Keep the query flexible.** Using a flexible query, like `sourceCategory=*Apache*` so that metadata changes don't break the query.
- **Include an aggregation.** Be sure to include a "group by" or aggregate function, and include any timeslice information in the aggregation; for example, the query could contain `timeslice by 1h | count by _timeslice`. Additionally use the smallest timeslice that makes sense, such as one hour instead of one day.
- **Include necessary fields, but avoid fields that tend to vary.** For example, you'd want to use country and city fields instead of latitude and longitude.

What types of operators aren't supported in Scheduled Views?

Scheduled Views are defined by a query, with the search results being indexed. Due to the way data is indexed in Views, several operators are not supported. Unsupported operators include:

- Accum
- CIDR
- Concat
- isNull
- Limit
- Lookup
- Matches
- Math operators

- Rollingstd
- Save
- Sessionize
- Smooth
- Summarize
- Total
- Trace
- Transpose
- URL Decode

Adding a Scheduled View

Any admin can delete or stop any View, but once a View has been set up it can't be edited.

To create a Scheduled View:

1. In the Sumo Logic Web Application, choose **Manage > Scheduled Views**.
2. Click **Create**.
3. In the Create a View dialog box, enter the following:
 - **View Name.** Type a name that you'll use to search the data in a query. It's important to use a name that's descriptive and easy to remember. Names can be comprised of alphanumeric characters; underscores(_) are the only special characters allowed.
 - **Query.** Type the full query that encompasses the data you'd liked indexed in the View. Parse operators and most search operators are supported in Views.
 - **Start Time.** Click the date that you'd like to use as the start time of the index. All data from that point forward will be indexed in the View. The oldest selectable date represents the end of the retention period of your Sumo Logic account.
4. Click **Create**.

The View begins to index data as soon as you create it. Allow a few hours for the indexing to complete. If you've chosen to index a large amount of data and/or have chosen a long date range for the view, it could take a bit longer.

To stop and restart a Scheduled View:

If you stop, then restart a Scheduled View, Sumo Logic starts aggregating from the time when it was stopped.

- Click **Stop** to the right of the View. Click **Start** to restart it.

Running a Search Against a Scheduled View

Running a search against the indexed data in a Scheduled View is almost exactly the same as running any other query. The difference you'll notice is the speed at which results are returned, especially if you're searching over a long period of historical data.

Queries that contain Views can be saved as scheduled searches, as Dashboard Monitors, and as published or saved searches.

To run a search against a Scheduled View:

1. Select **Manage > Scheduled Views**, then copy the name of the View from the Scheduled View page.
2. Click **Search** in the top ribbon to open the Search page.
3. In the search text box, type `_view=`, then paste or type the name of the View, like: `_view=[viewName]`.
4. Complete the query, or run the search just on the View.

Using Timeslices in Scheduled Views

A Scheduled View indexes data every minute, meaning that if you have a one hour timeslice, a search would return 60 results.

When defining a Scheduled View, you can choose to include a timeslice to segment the data. It's important, however, to note that data indexed into a Scheduled View is somewhat segmented already because a Scheduled View adds new data every minute. In other words, if you have a one hour timeslice running against a Scheduled View, you'd get 60

What happens if you use a 15-min timeslice considering data is indexed in one-min increments?

Example of re-timeslicing. Why??

When running a query against a view, you can choose to include another timeslice in the query to re-order the data.

Running a Search Against a Scheduled View

Running a search against the indexed data in a Scheduled View is almost exactly the same as running any other query. The difference you'll notice is the speed at which results are returned, especially if you're searching over a long period of historical data.

Queries that contain Views can be saved as scheduled searches, as Dashboard Monitors, and as published or saved searches.

To run a search against a Scheduled View:

1. Select **Manage > Scheduled Views**, then copy the name of the View from the Scheduled View page.
2. Click **Search** in the top ribbon to open the Search page.
3. In the search text box, type `_view=`, then paste or type the name of the View, like: `_view=[viewName]`.
4. Complete the query, or run the search just on the View.

Data Volume Index

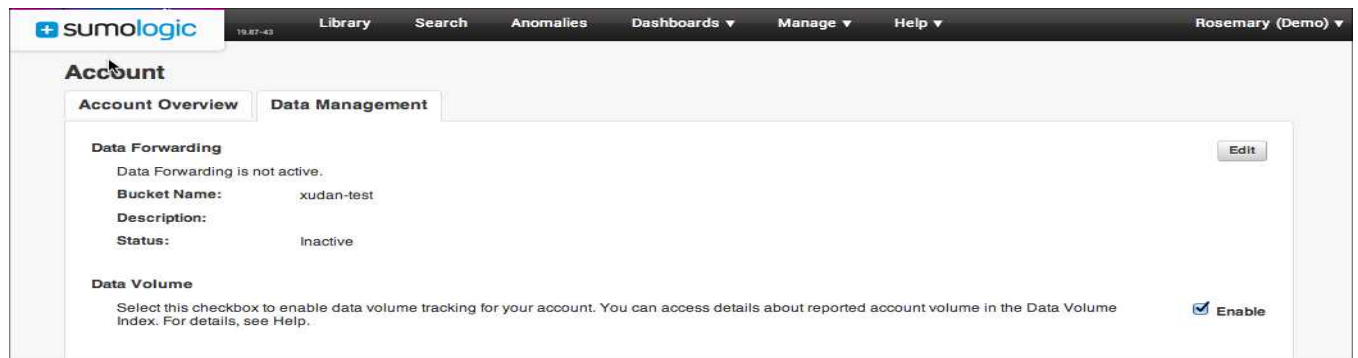
The Data Volume Index automatically provides data that allows you to understand your account's data ingest volume in bytes and number of log messages processed overall. The Data Volume Index gives you better visibility into how much data you are sending to Sumo Logic, allowing you to proactively manage your systems' behavior and to fine tune your data ingest with respect to the data plan for your Sumo Logic subscription.

Before it can be used, the Data Volume Index must be manually enabled by an administrator. Once enabled, it will begin populating, and create a set of log messages within the Data Volume Index every five minutes. It's important to note that data does not backfill. Also, data is only provided to the Data Volume Index while the option is enabled.

Once enabled, you can access the Data Volume Index using the search query `_index=sumologic_volume`.

Enable the Data Volume Index

Before it can be used, the Data Volume Index must be enabled by an administrator.



To enable the Data Volume Index:

1. Go to **Manage > Account**.
2. Select the **Data Management** tab.
3. Under **Data Volume**, select **Enable**.

A message confirms that the feature is enabled.

Using the Data Volume Index

The Data Volume Index is populated with a set of log messages every five minutes, which contains information on how much data (by bytes and messages count) your account is ingesting. Each log message includes information based on one of the following Index Source Categories.

Index Log Type	Index Source Category
Collector	collector_volume
Source	source_volume
SourceName	sourcename_volume

SourceCategory	sourcecategory_volume
SourceHost	sourcehost_volume
View	view_volume

You can query the Data Volume Index just like any other message using the Sumo Logic Search page. To see the data created within the Data Volume Index, when you search, specify the `_index` metadata field with a value of `sumologic_volume`.



For more information see [How metadata is used in searches](#).

To query the Data Volume Index:

1. In the Search page, enter the query `_index=sumologic_volume`.
Important: Make sure to enter the query exactly as shown to search against this specific source.
2. Choose the time range for the data that you'd like to review.
3. Click **Start** to run the search.

Results are returned in the **Messages** tab.

To further limit the search results to the Data Volume Index data for a specific volume category, you can supply the Index Source Category using the `_sourceCategory` metadata and one of the Index Source Categories from the previous table. For example:

`_index=sumologic_volume AND sourceCategory=collector_volume`



If the feature is not enabled, a search will not produce any results.

Data Volume Index Message Format

The Data Volume Index messages are JSON formatted messages that contain parent objects for each source data point, and child objects that detail the message size and count for each parent.

For example, a single message for the "Collector" volume data may look similar to the following, with `collector_X` representing the Collector names. The `sizeInBytes` and `count` values are the aggregated volume for that five minute time period.

```
{
  "collector_a":{"sizeInBytes":733296,"count":1646},
  "collector_b":{"sizeInBytes":4380031,"count":12105},
  "collector_c":{"sizeInBytes":386255,"count":843},
  "collector_d":{"sizeInBytes":10823082,"count":23923},
  .
}
```

```
.  
}
```

Examples

Volume for Each Category

This example query will return the volume for each Source Category.

```
_index=sumologic_volume _sourceCategory=sourcecategory_volume  
| parse regex "\"(?<sourcecategory>(?:[^\"]+))\"(?:\"sizeInBytes\"(?:<bytes>\d+),\"count\"(?:<count>\d+))\" multi  
| bytes/1024/1024/1024 as gbytes  
| sum(gbytes) as gbytes by sourcecategory
```

would produce results such as:

Messages		
Aggregates		
Page: 1 of 4		
#	sourcecategory	gbytes
1	analytics_metrics_reporter	3.35515e-4
2	collector	0.02774
3	raw	0.00148

Volume for Each Collector

This example query will return the volume for each Collector.

```
_index=sumologic_volume _sourceCategory=collector_volume  
| parse regex "\"(?<collector>(?:[^\"]+))\"(?:\"sizeInBytes\"(?:<bytes>\d+),\"count\"(?:<count>\d+))\" multi  
| bytes/1024/1024/1024 as gbytes  
| sum(gbytes) as gbytes by collector
```

would produce results such as:

Messages		Aggregates	
<div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div>		Page: 1	of 8
#	collector	gbytes	
1	long-index-18	0.00949	
2	long-index-30	0.00804	

Volume for a Specific Source

The following query returns the message volume for a specific Source. The Source name can be supplied within a JSON operation to get the child objects for that Source.

```
_index=sumologic_volume _sourceCategory=source_volume
| json "my_source_name" as source
| json field=source "sizeInBytes" "count"
| sizeinbytes/1024/1024/1024 as gbytes
```

Volume for a Specific Collector

The following query returns the message volume for a specific Collector. The Collector name can be supplied within a JSON operation to get the child objects for that Collector.

```
_index=sumologic_volume _sourceCategory=collector_volume prod-receiver-10
| json "prod-receiver-10" as collector_json
| json field=collector_json "sizeInBytes", "count" as bytes, count
| sum(bytes) as bytes
| bytes/1024/1024/1024 as gbytes
| fields gbytes
```

Sumo Logic App for Data Volume

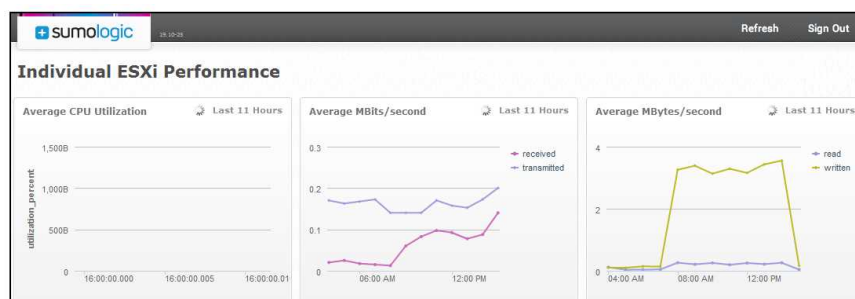
Sumo Logic provides an application that utilizes the Data Volume Index to see your account's volume usage as a glance. For details, see [Data Volume app.htm](#).

About Dashboards

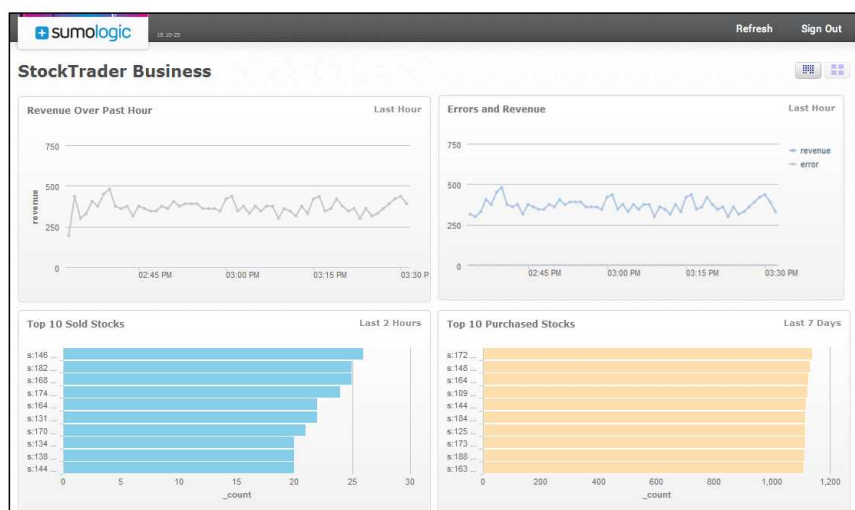
Dashboards contain a collection of real time **Monitors** that provide a graphical representation of your organization's data. The information you save in a Dashboard provides insight into the current state of the data you're uploading to Sumo Logic. Instead of having to run a number of queries, Sumo Logic runs these searches automatically, making sure that you're never looking at stale data. You won't have to remember to run important searches once you save them to Dashboards—you'll be able to spend your time gaining insight into what's happening in your organization.

The uses for Dashboards are nearly endless. Perhaps your IT security group wants to keep an eye on who is installing virtual machines. Save the query you run as a Monitor, and watch for spikes in a line graph. Dashboards bring additional assurance, knowing that unusual activity will be displayed real time in an easy-to-digest graphical format. The data that matters the most to you is even easier to track.

You can configure each Dashboard to display information for different operational responsibilities in your organization. So in one Dashboard you can keep an eye on 404 errors or other website issues, while another Dashboard focuses on security policies. Or, for example, you can keep an eye on a particular server so you're aware of the condition of your machine:



Perhaps you handle financial information in addition to IT data. You could create a Dashboard focusing on what's happening in your business:



What else do I need to know before I get started?

Here are a few things to keep in mind about Dashboards.

Dashboards can be published to leverage the data you've analyzed.

You've saved Monitors to a Dashboard, and you'd like to enlighten others in your organization. Just publish the Dashboard to allow everyone to see what you've uncovered through your analysis.

Dashboards are built to your specifications.

As long as you run a search with aggregate results, you can save that search in a Dashboard as a Monitor. For example, you could create a Dashboard filled with performance-based searches, and another to track session length or other unique searches. Whatever makes sense for you and your organization—create a Dashboard and you're set.

You can also choose the number of columns in a Dashboard, allowing you to view data in the most compelling format for your needs.

[Learn more about setting up Dashboards](#)

Dashboards may take some time to fill.

Depending on the time range of a Monitor, it can take time for a Monitor to display the complete results. Monitors are built as data is ingested. So if you create a one-hour monitor, it'll take an hour to see the full results; for a 30-day Monitor, it will take 30 days.

Dashboards require queries that have aggregate results.

Only searches that produce aggregate results can be saved as Monitors. What does that mean? Basically, to produce aggregate results, your query will contain a [grouping](#) function. This allows the data to be displayed properly in a graphical format.

Dashboards can't support all queries.

Certain operators cannot be used in Dashboard Monitors. A list of unsupported operators can be found in [Adding Monitors](#).

Using Dashboards in Sumo Logic Free Accounts

Dashboards in Sumo Logic Free accounts work the same way as those in Enterprise accounts with one limitation: organizations with Sumo Logic Free accounts have access to 20 Monitors, shared between the three users in the account. You can decide to build as many Dashboards as you'd like with the available Monitors in your account.

Can I share Dashboards in a Sumo Logic Free account?

Yes. Sharing Dashboards works the same in Sumo Logic Free and Enterprise accounts. Once you share a Dashboard it's made available to all the users in the account.

Adding Monitors to a Dashboard

Almost any query you run that produces aggregate results can be saved as a **Monitor** in a Dashboard.

Aggregating (grouping) functions evaluate messages and place them into groups. After a search completes, you can choose a layout for the Monitor (although you'll be able to change the layout of the Monitor at any time), then either add the Monitor to an existing Dashboard or create a new Dashboard from the Monitor.



If you're creating your first Dashboard, the steps are the same as adding a Monitor to an existing Dashboard. You'll first run a search, add the results as a Monitor, then name and create the new Dashboard.

To add a Monitor to a Dashboard:

1. After running a query, choose the layout for the Monitor from the chart icons (table, bar chart, column chart, line graph, area chart, pie chart, or map) in the Aggregates tab. (You'll be able to change the layout of the Monitor at any time.)



2. Click the **Add to Dashboard** icon in the Aggregates tab.



3. In the Add to Dashboard dialog box, do the following, then click **Add**:
 - For **Dashboard**, choose the name of the Dashboard to which you'd like to add the Monitor. If you'd like to create a new Dashboard, type a name in the Dashboard dialog box.
 - For **Column**, choose an existing column where you'd like to display the Monitor, or to create a new column, type a name in the Column text box. (New columns are always added to the right of existing columns.)
 - For **Title**, type the name of the Monitor.



Add to Dashboard

Dashboard Errors
Select a Dashboard or create a new one by entering its name.

Column By Source
Select a Column or create a new one by entering its name.

Title Errors (15 minutes)
Type a title for the new Monitor.

Cancel | **Add**

The Dashboard displays the new Monitor.

Why does the data look different in the new Monitor?

The results of a query in the Search tab of the web application are displayed based on the time range you chose when you initially ran the search. When you add the search as a Monitor, the query is essentially re-run using a fresh time range. If you've just added a Monitor with a long time range, like 24 hours, it can take up to 24 hours to see the full results in a Monitor.

Restricted Operators

In some cases searches can't be added to a Dashboard. Queries using the following operators can't be saved as Monitors:

- CountFrequent
- Details
- Save
- Sessionize
- Summarize
- Trace

The following Operators can be used in Dashboard Monitors, but in the search they must be included *after* the first "group-by" phrase:

- Limit
- SortBy
- Top

Publishing Dashboards

Publishing a Dashboard is a great way to keep everyone on top of data that is important to your organization. After a Dashboard is published others can subscribe to it, which means that any changes you make will also be reflected in the subscribers' accounts.

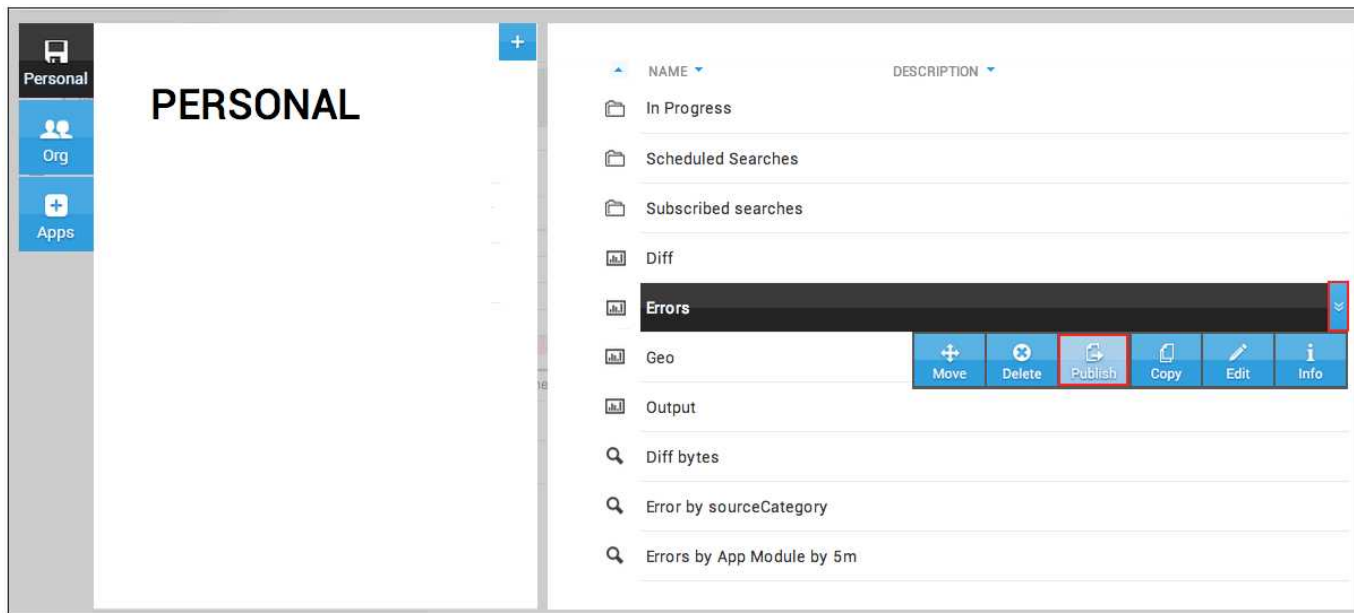
Once you publish a Dashboard, a user can **Subscribe** to it, meaning that as you make changes to it, those changes are reflected in the other user's Library as well. Think of a published Dashboard as being "pushed" to a user who subscribes; any changes are pushed (updated) to subscribers.



Others who view your published Dashboards can run queries based on Monitors you've saved in the Dashboard. Then they can save their own versions.

Publishing Dashboards from the Library

1. Click a Dashboard, then click the double arrow to the right of the name. Click **Publish**.



2. When prompted to confirm the change, click **Publish**. The search is moved to your Org folder:

To unpublish a Dashboard:

1. In your **Org** folder, select a published Dashboard, then click the double arrow to the right of the name. Click **Unpublish**.
2. When prompted, click **OK**.

The content is moved to your Personal folder.

Publishing from the Dashboards page

In addition to using the Library, you can choose to publish (or unpublish) a Dashboard directly from the Dashboard menu.

To publish a Dashboard:

1. Click the Dashboard's **Properties** icon, located right next to the Dashboard's name.
2. Choose **Publish Dashboard**.



To unpublish a Dashboard:

1. Click the Dashboard's **Properties** icon.
2. Choose **Unpublish Dashboard**.



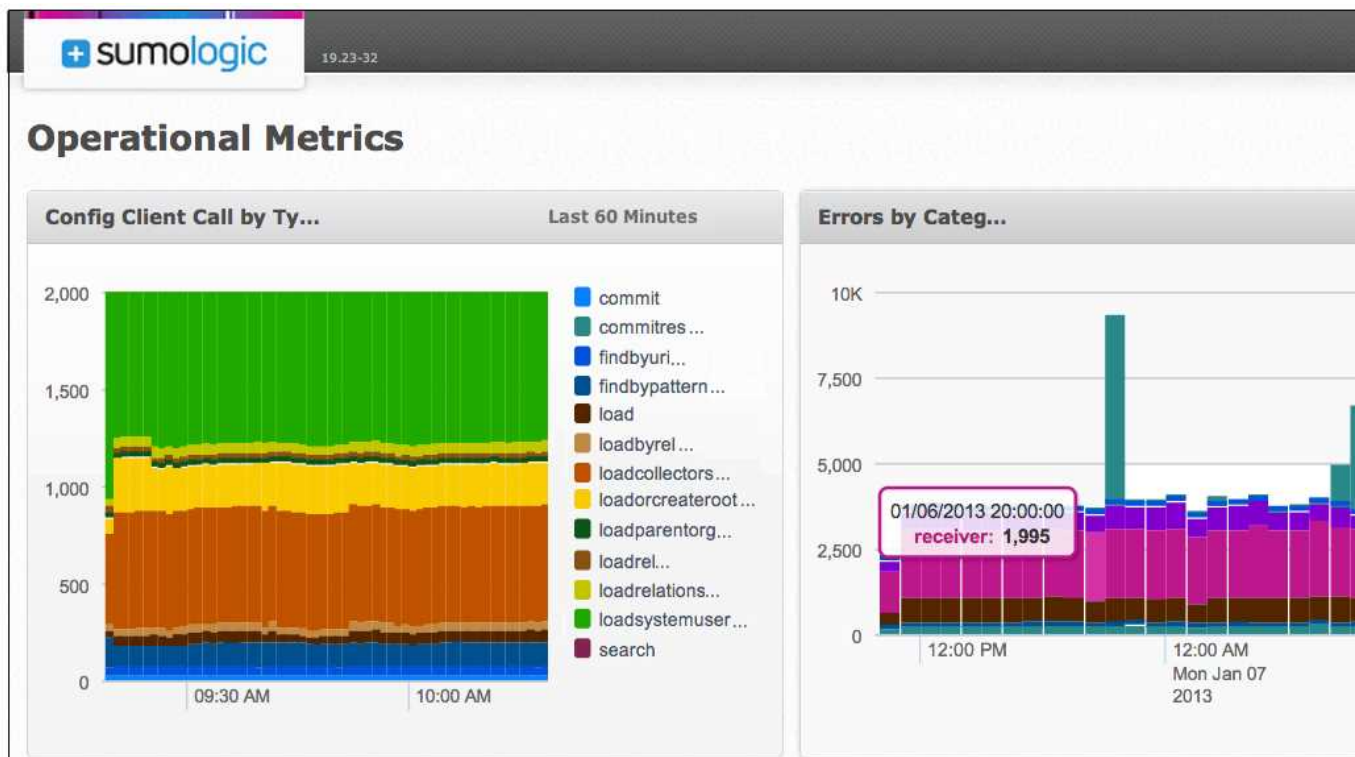
Launching a search from a Monitor

Whether you're working with a Dashboard you designed or from a shared Dashboard, you can click a data point in a Monitor to launch a search from that point in time. This is especially useful in situations where you've noticed activity in a Monitor that you'd like to research further.

After running the search you can save the query as a new Monitor, or save any changes you've made to the search back to the existing Monitor.

To launch a search from a Monitor:

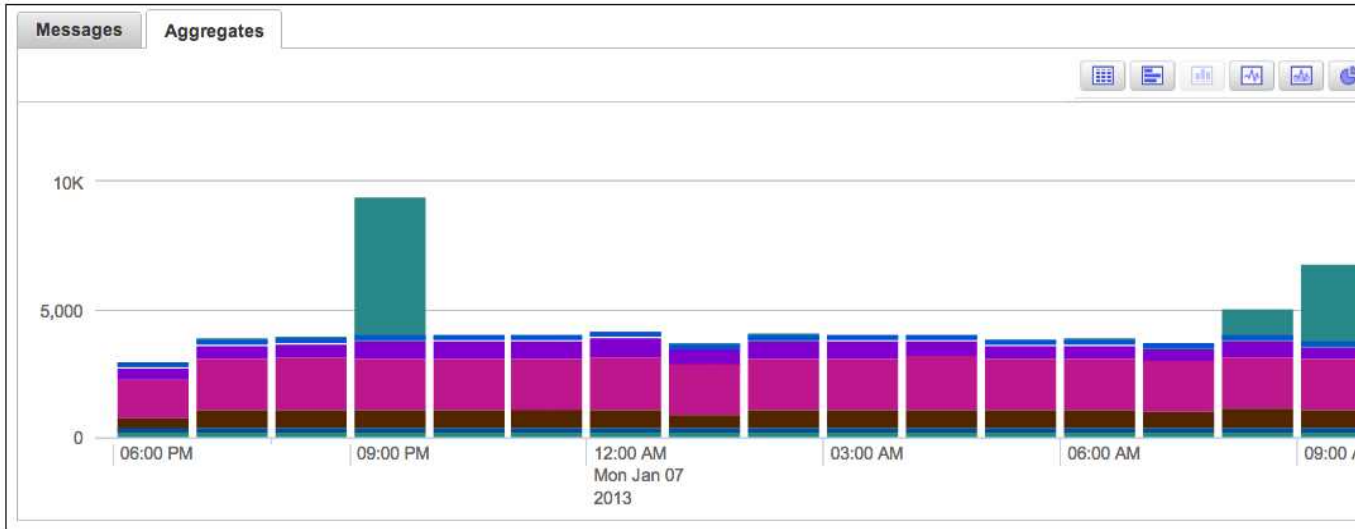
1. Click a bar of a bar or column chart, or a data point in a line chart.



2. Click **OK** to confirm the search.



The search begins. Results are displayed in the Aggregates tab.



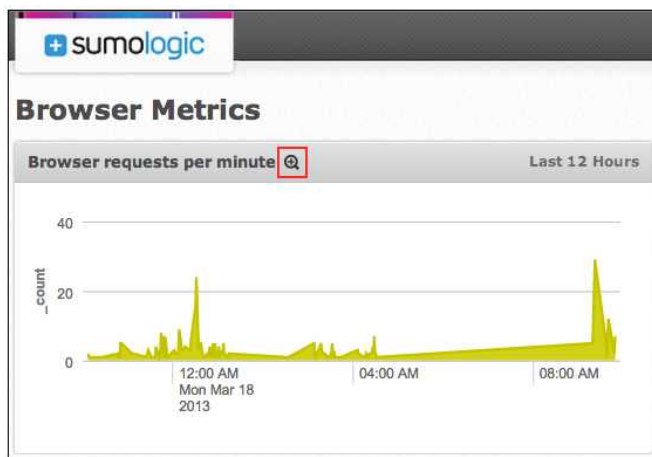
Zooming in on a Monitor

The Zoom icon allows you to view a single Monitor. The zoomed Monitor fills the browser window, so you control the size of the Monitor.

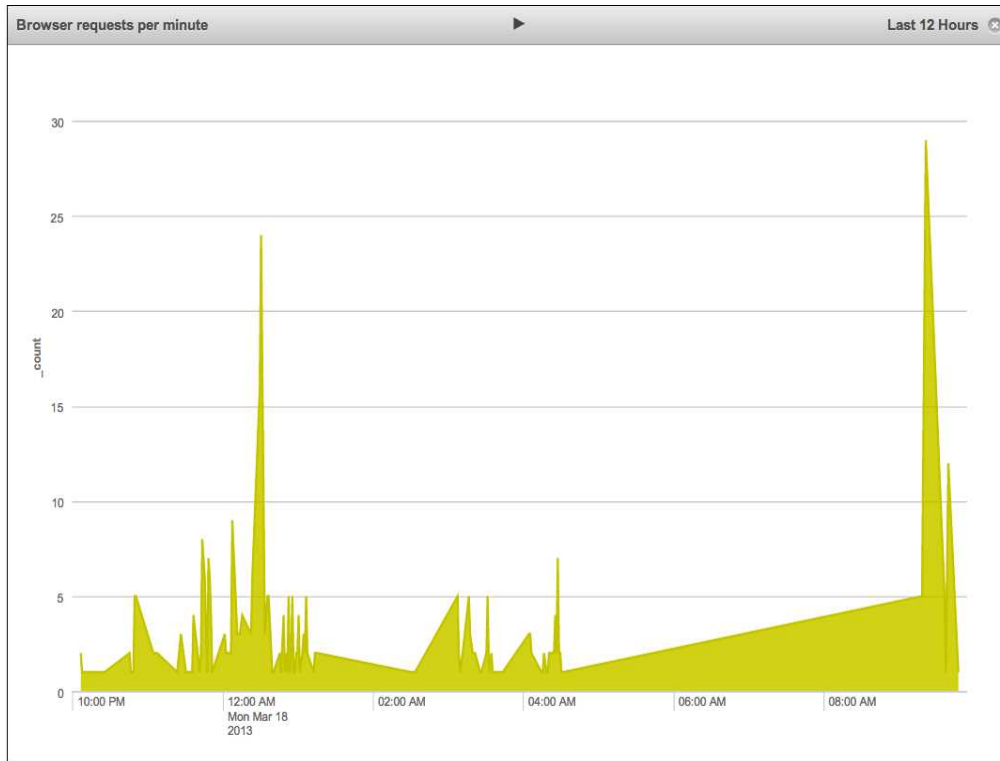
While viewing a zoomed Monitor, you can also "pause" the data streaming in to the Monitor. Instead of viewing real time search results, you can get a static view. When you're ready, you can to return to real time with the click of a button.

To zoom in on a Monitor:

- In a Dashboard (either shared or owned by yourself) hover over the name of a Monitor, then click the zoom icon:



The Monitor is displayed at a larger size:

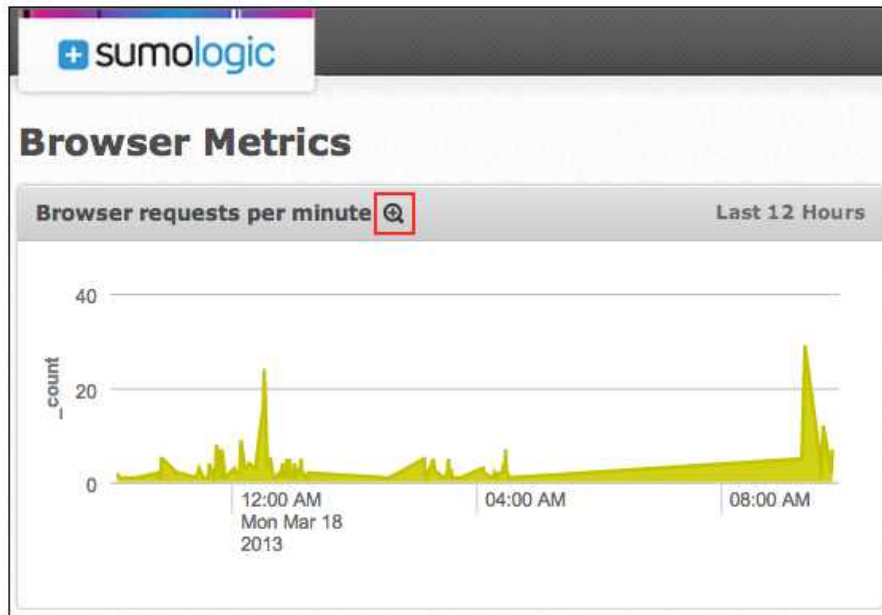




Pausing a Monitor

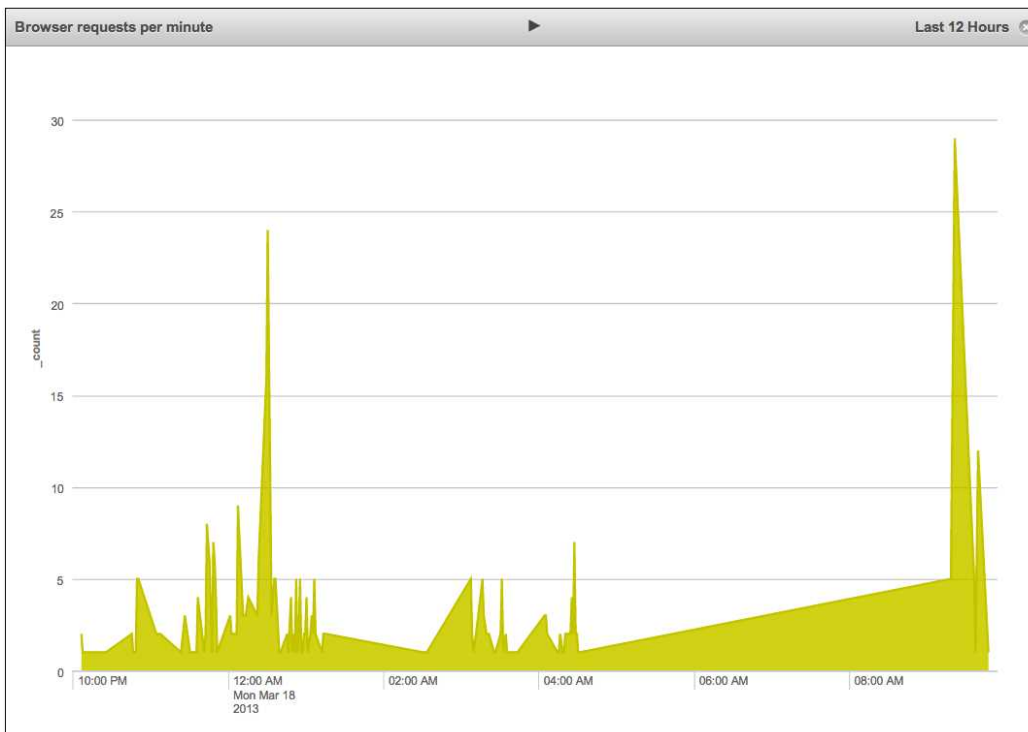
Because Monitors display real time data, the data displayed is constantly updating. If you'd like to view static data, you can "pause" a Monitor when it's zoomed to temporarily disable auto updates.

To pause a Monitor:

1. In a Dashboard (either shared or owned by yourself) hover over the name of a Monitor, then click the zoom icon:



2. Click the Play icon ; it's replaced with the Pause icon .
3. When you're ready to re-enable auto updates, click the Pause icon again. When the play icon displays, auto updates have been re-enabled:



Editing Dashboards and Monitors

Several Monitor properties can be edited directly from a Dashboard. (If you'd like to edit the query saved as a Monitor, see [Saving changes back to a Monitor](#).) When editing a Dashboard or Monitor, you'll use the **Properties** menu:



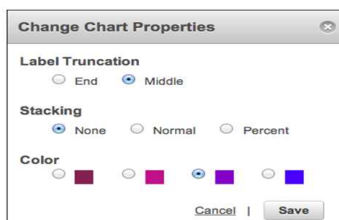
- A. Dashboard Properties menu.
- B. Monitor Properties menu.

Changing Monitor chart properties

The colors used in a Monitor are assigned randomly. You can change the color at any time by editing the Chart Properties. You'll also find the option to change the way labels are truncated.

To change Chart Properties of a Monitor:

1. Click the Monitor's **Properties** icon, and then choose **Change Chart Properties**.
2. Do any of the following:
 - To change the way labels are truncated, choose either **End** or **Middle**. If you choose **End** label truncation, it means that a value, such as an IP address, could be shortened to **100.100....**, so that the beginning of the value is retained. Choosing **Middle** means that an IP address could be shortened to **100....100**, so that the beginning and the end of the value is retained.
 - For stacked charts, choose an option for **Stacking**. (This option won't change the appearance of any other type of Monitor.)
 - To change the color scheme used in the Monitor, choose a color option.
3. Click **Save**.

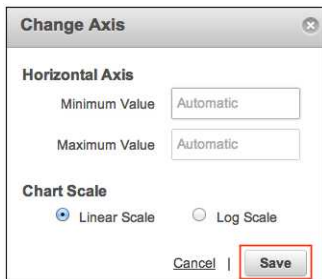


Changing the axis of a Monitor

You can change the minimum and maximum values of a Monitor to influence the results in a Monitor. You can also set the Monitor to use Log Scale.

To change the axis of a Monitor:

1. Click the Monitor's **Properties** icon, and then choose **Change Axis**.
2. Do any of the following:
 - To edit the default (automatic) values used for a Monitor's axis, type specific values for **Minimum Value** and **Maximum Value**.
 - To change to either **Linear Scale** (default) or **Log Scale**, choose that option.
3. Click **Save**.

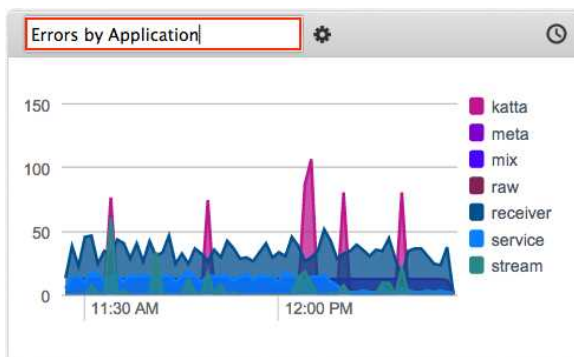


Changing the name of a Dashboard or Monitor

The names of Dashboard and Monitors can be changed at any time.

To change the name of a Dashboard or Monitor:

1. Click the name of the Dashboard or Monitor. Type a new name when the text box appears.



2. Press **Return** or **Enter**.

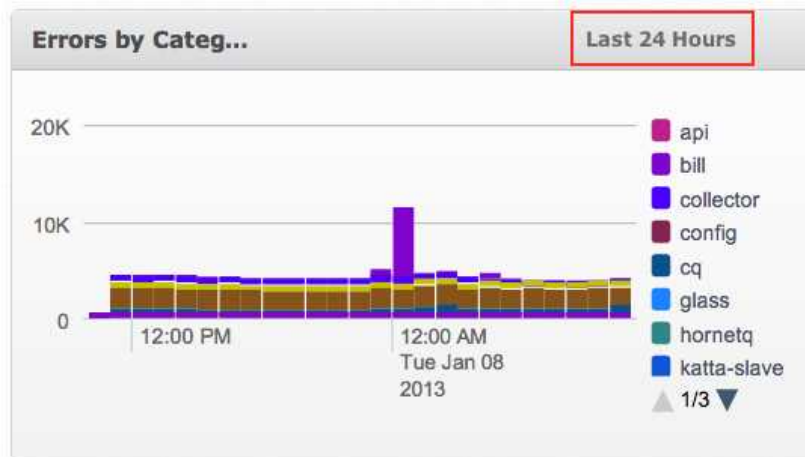


If you click away from the text box, the name won't be changed.

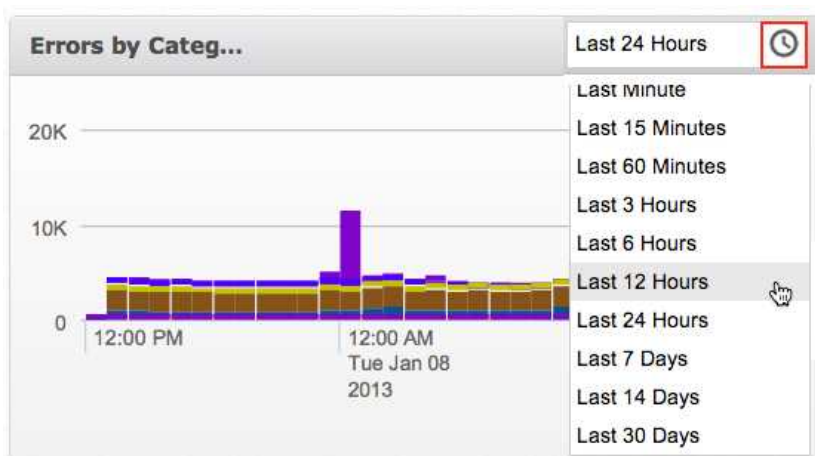
Changing the time range of a Monitor

Once you've created a Monitor, you can change the time range whenever you'd like.

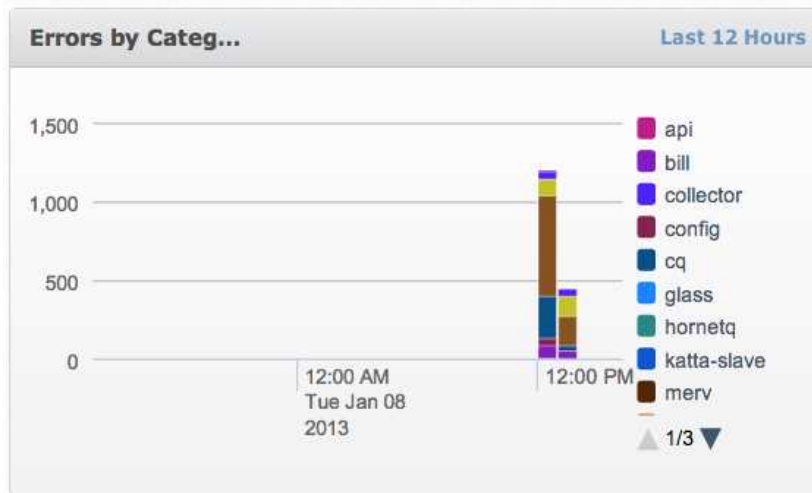
1. Click the Monitor's time range.



2. Click the time range once again to display a menu. Choose an option to change the Monitor's time range.



The Monitor's time range is reset, and the Monitor's query is re-run using the new time range. If you've chosen a longer time range, it may take time to see complete results. This is especially true for queries with long time ranges. (It will take 12 hours for full search results to display if you're running a 12-hour query.)



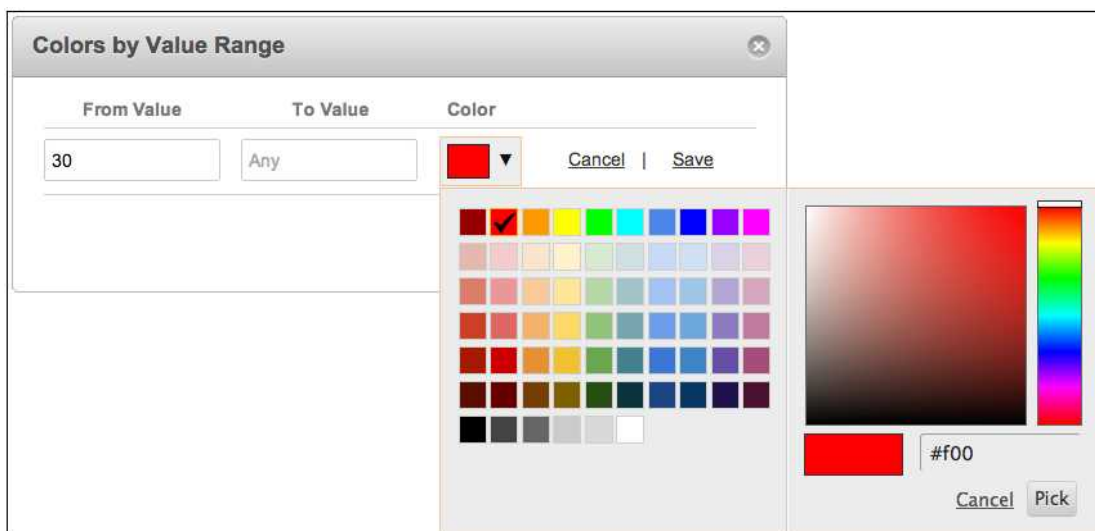
Changing the color of a bar or column by value range

When viewing search results displayed as a bar chart or column chart in Dashboard Monitors or in the **Search** tab, you can choose to change the color of a bar or column depending on the value range of the search output. This setting doesn't affect the query; it's only applied to query results.

At a glance, you'll be better able to see if a value is out of an acceptable range, or is under an important threshold. For example, say we've created a Dashboard Monitor that displays results for user log ins over 24 hours. We'd like to see right away any user that has logged in more than 30 times in the past day. We can have a column that shows 30 or more log ins display as bright red, which is easily distinguishable from our other columns.

To change the color of a column by value range:

1. In a bar chart or column chart Monitor, click the **Properties** icon and choose **Colors by Value Range**.
2. In the **Colors by Value Range** dialog box, type a number in the **From Value** text box. You can leave the **To Value** text box empty if you don't want to set an explicit value; otherwise type a number in the box.
3. For **Color**, choose a pre-set option, or choose a customer color.



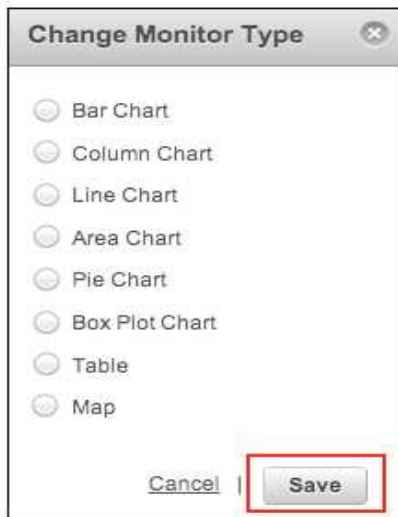
4. Click **Save**.

Changing the type of a Monitor

If you'd like to change the way a Monitor is displayed in a Dashboard you can choose a different layout.

To change a Monitor's type:

1. Click the Monitor's **Properties** icon, and then choose **Change Monitor Type**.
2. Select a different layout, and then click **Save**.



Saving edits to a Monitor

After running a search from a Monitor, any edits made to the query can be saved back to the Monitor. These changes are immediately reflected in the Monitor. In addition to fine-tuning the query, you can also change the time range of the Monitor.



Only the owner of a shared Dashboard can save changes back to a Monitor.

To save changes back to a Monitor:

1. Click a Monitor to launch a search.
2. Make any changes you'd like, either to the query or the time range of the Monitor.
3. Run the search.
4. Once the search has completed, click the **Save Back to Monitor** icon.



5. Confirm the changes you've made.

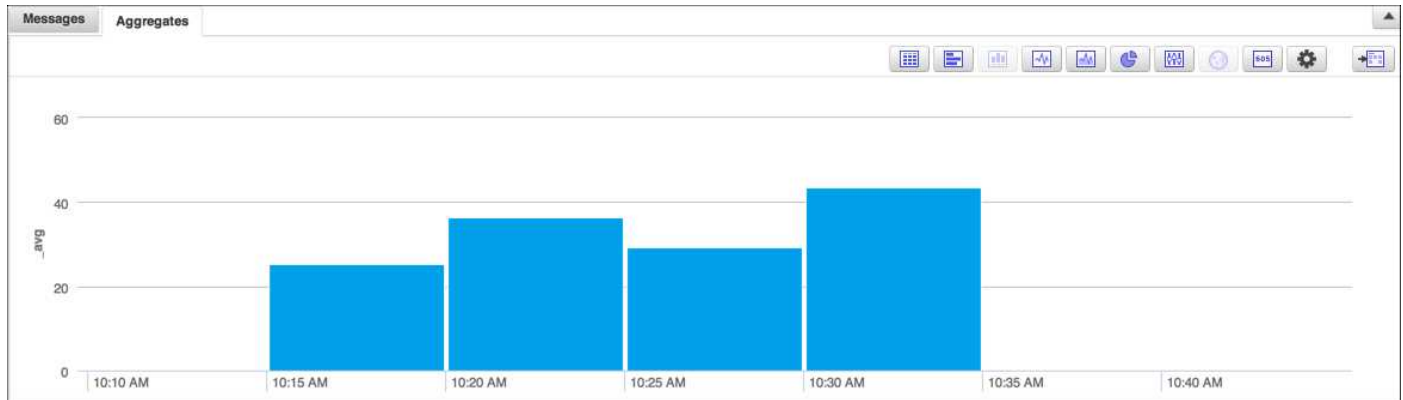
Changes are reflected immediately.

Creating combo chart Monitors

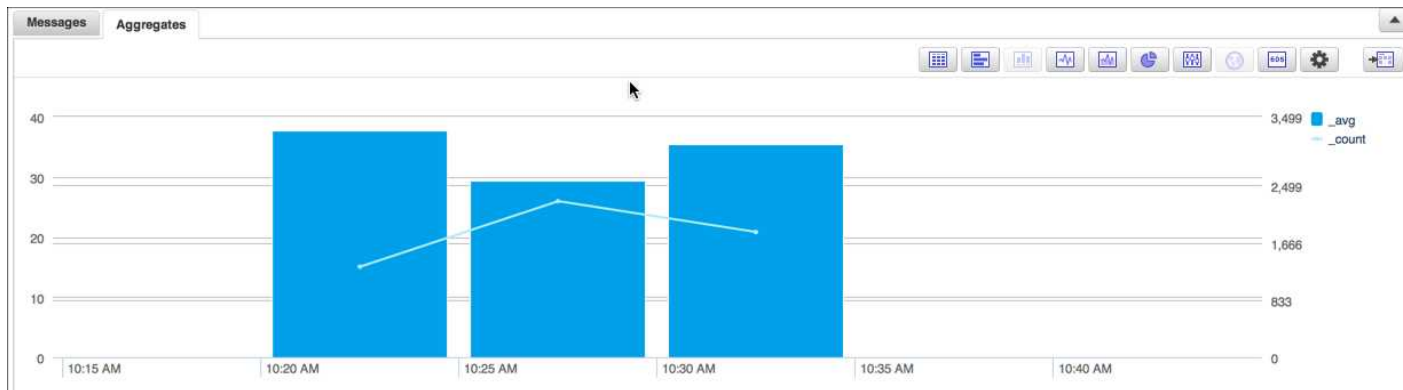
Combo charts allow you to create bar or column charts with a secondary vertical axis so you can add a separate line as a comparison. For example, say we're running a query like:

```
_sourceCategory=*apache* | parse "HTTP/1.1\" * * * " as status_code, size, response_time | timeslice  
5m | avg(response_time), count by _timeslice
```

Generally, this query would produce a bar or column chart that looks a bit like this:



But with a combo chart, we can set the `_count` volume to display as a line chart on a second axis, so the count value is represented by a line instead of a bar:



Then you can add combo charts to a Dashboard or view them in the **Aggregates** pane.

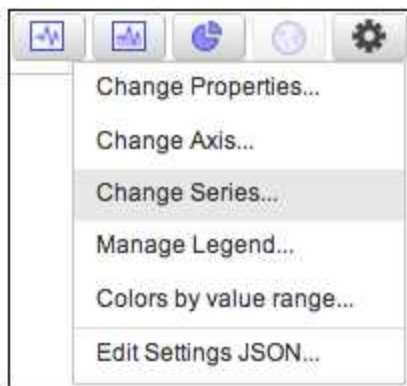
To create a combo chart:

1. Run a query.

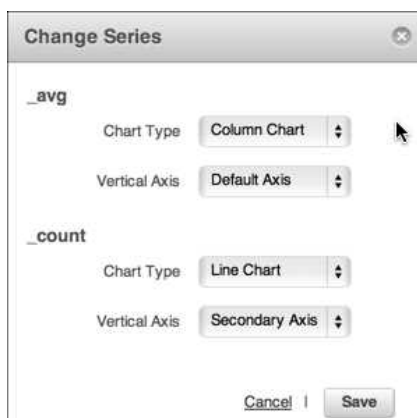
2. In the **Aggregates** tab, choose **column chart** or **line chart** to display the search results.



3. To add a second axis, click the **properties** icon in the **Monitor** or **Aggregate** pane, then choose **Change Series**.



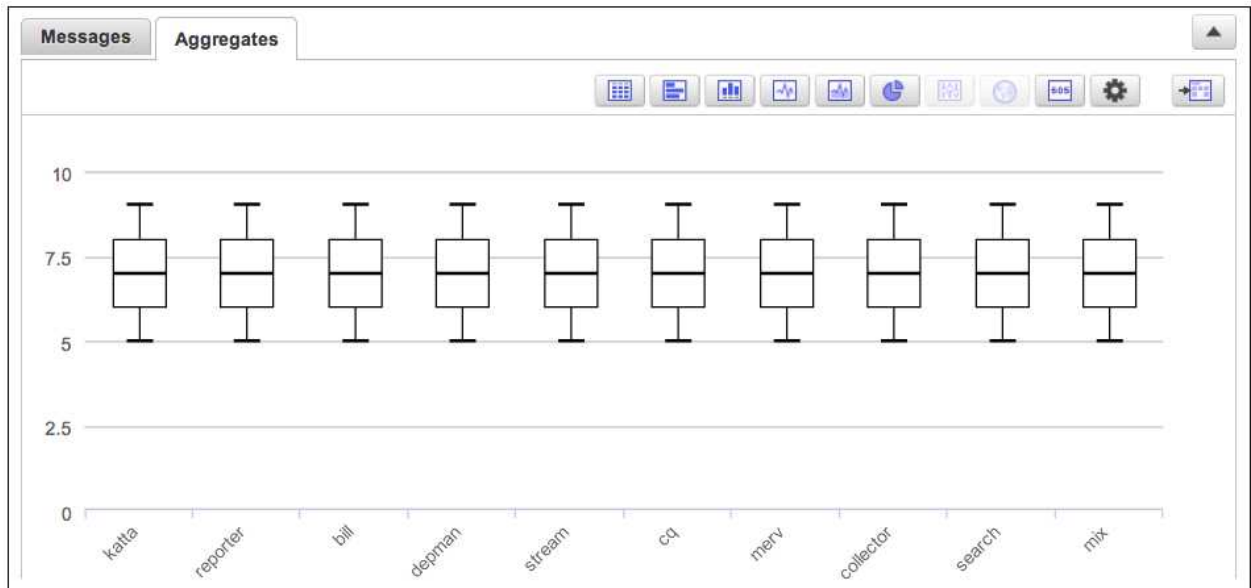
4. In the **Change Series** dialog box, choose the value you'd like to add as a secondary axis and chart type. In our example, we set the **_count** value as a **Line Chart** on the **Secondary vertical axis**. Click **Save**.



The combo chart is built and displayed.

Creating Box Plot Charts

A Box Plot Chart graphically depicts groups of data using their quartiles. In Box Plot charts, the bottom and top of the box represent the first and third quartiles; the band inside the box represents the median:



To create Box Plot Chart Monitors, your query must include the:

- **Smallest value** (sample minimum) using the `min` or `_min` field name.
- **Lowest quartile** (25%) using the `_pct_25` field name. You can use both `lower` or `ends` with in this part of the query.
- **Median quartile** (50%) using the `_pct_50` field name. You can use both `lower` or `ends` with in this part of the query.
- **Upper quartile** (75%) using the `_pct_75` field name.
- **Largest value** (sample maximum) using the `max` or `_max` field name.

For example, this query can be rendered as a Box Plot Chart:

```
error| 5 as a | 6 as b | 7 as c | 8 as d | 9 as e | min(a), pct(b,25), pct(c,50), pct(d,75), max(e)
```

But, because this query doesn't meet all the requirements, it cannot be rendered as a Box Plot Chart:

```
error | 5 as a | 7 as b | 7 as c | 7 as d | avg(a, b), max (c,d), min(c)
```

The above query is missing the lower, median, and upper quartile values.

To create a Box Plot Chart:

1. Type a supported query in the Search box, making sure to include all of the required field names.
2. Once the search results appear, click the **Box Plot Chart** icon.



3. (Optional) Click **Add to Dashboard** if you'd like to save the chart as a Monitor.

Creating Single Value Charts

A Single Value chart is useful for displaying the results of a query that returns only a single value or record, in order to make that value stand out at a glance. If the query returns more than one value in the **Aggregation** tab, only the first value is displayed in the Single Value chart.

For example, you could use a Single Value chart in order to alert you to whether a system is up or down, or to give a “stop-light” alert status of a single value and when it changes. The Single Value chart is especially useful for security analytics and PCI (Payment Card Industry) compliance, in order to display the number of incidents under specific regulations.



If your query does not meet the requirements for displaying a Single Value chart, for example, if there are no results or if it is a null result, the Single Value icon will appear greyed out in the display options.

You can create a Single Value chart for three query result options: numerical, string, or Boolean.

Numerical Single Value Chart

A numerical Single Value chart displays an important metric for single glance analysis, for example, for the number of errors generated by your site.

This simple query provides a single value as a result:

error | count

for example:

Messages		Aggregates	
Page: 1		of 1	
#	_count		
1	27,649		

If your query provides more than a single result, when you create a Single Value chart, only the first result from the table in the Aggregates tab is displayed.

To create a Numerical Single Value Chart:

1. Run a query for a single value.
2. In the **Aggregates** tab, choose the **Single Value** icon to display the results graphically.



The Single Value chart is displayed, using default colors and text.

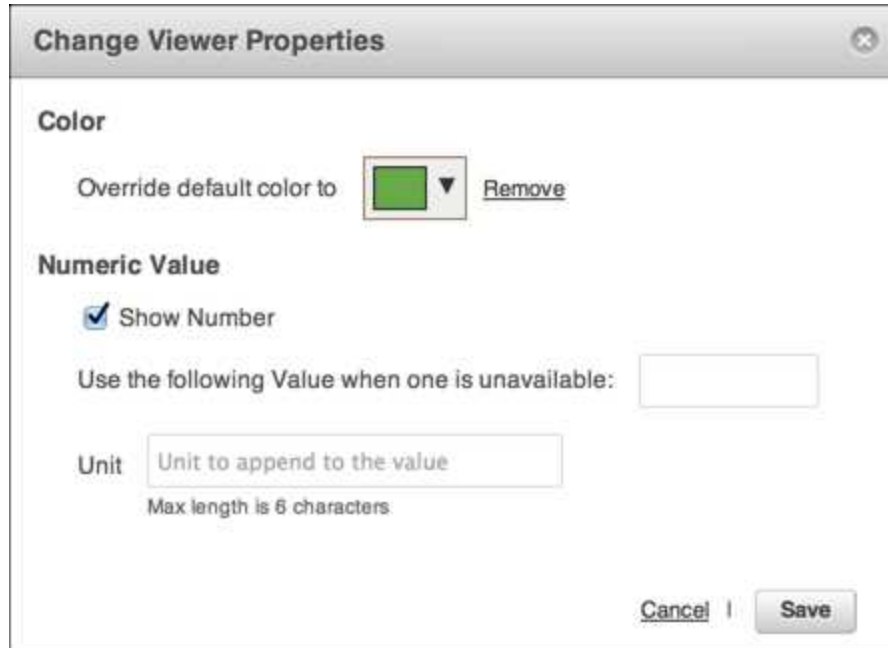


3. (Optional) Click **Add to Dashboard** if you'd like to save the chart as a Monitor.

To modify a Numerical Single Value Chart:

1. Select the **Properties** icon.
2. Select **Change Properties**. Here you can change:
 - **Color**. Override the default color to a new color.
 - **Numeric Value**. Here you can show or hide a numeric value, define a value to display where one is

unavailable, and define the unit to append to the value. Click **Save** to save changes.





The **Change Viewer Properties** dialog box has a close button (X) in the top right corner. It contains two main sections: **Color** and **Numeric Value**. In the **Color** section, there is a label "Override default color to" followed by a green color swatch with a dropdown arrow and a **Remove** link. The **Numeric Value** section includes a checked checkbox for **Show Number**, a label "Use the following Value when one is unavailable:" followed by an empty text input field, and a **Unit** label followed by another empty text input field containing the placeholder text "Unit to append to the value". Below the unit input field is the text "Max length is 6 characters". At the bottom right are **Cancel** and **Save** buttons.

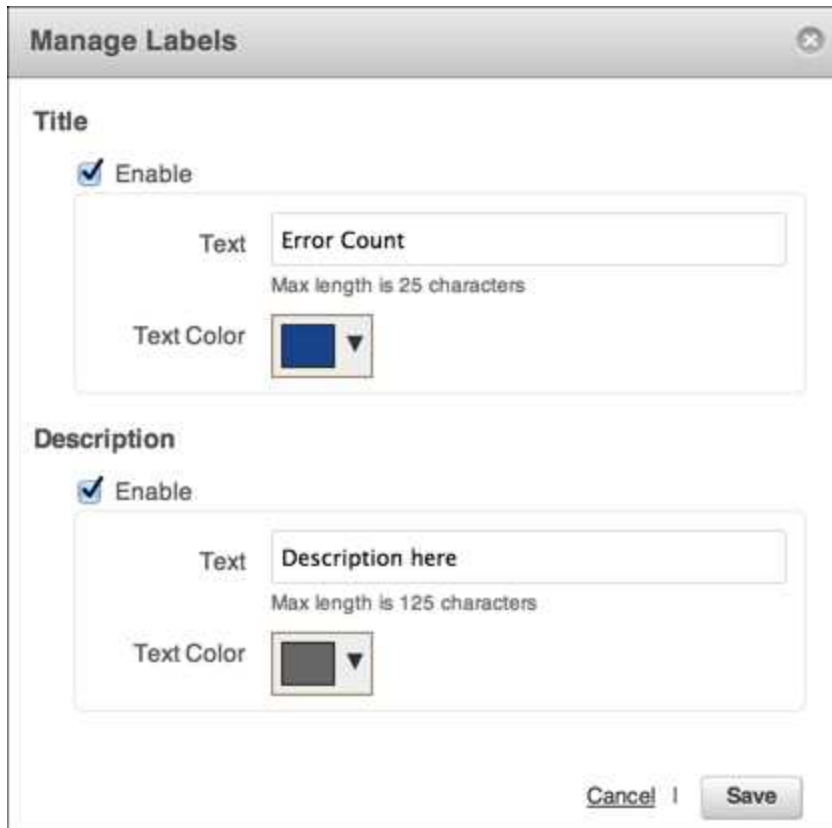
3. **Colors by Value Range.** Add colors by a value range. For example, you could choose a value from 0 to 1 as green, to show no errors. Then click **Add**, and add a value from 1 to Any as red, to show that errors have occurred. Click **Save** to save changes.



The **Colors by Value Range** dialog box has a close button (X) in the top right corner. It features a table with three columns: **From Value**, **To Value**, and **Color**. The table contains two rows: the first row has "0" for From Value, "1" for To Value, and a green color swatch; the second row has "1" for From Value, "Any" for To Value, and a red color swatch. To the right of each color swatch are **Edit** and **Delete** links. Below the table is an **Add** button. At the bottom right are **Cancel** and **Save** buttons.

From Value	To Value	Color	
0	1		Edit Delete
1	Any		Edit Delete

4. **Manage Labels.** This option allows you to enable a **Title** and a **Description**, including the text displayed and the color of the text. Click **Save** to save changes.



The 'Manage Labels' dialog box contains two sections: 'Title' and 'Description'. Each section has an 'Enable' checkbox, a 'Text' input field, and a 'Text Color' color picker. The 'Title' section has 'Error Count' in the text field and a blue color picker. The 'Description' section has 'Description here' in the text field and a grey color picker. Both text fields have character limits: 25 for Title and 125 for Description. 'Cancel' and 'Save' buttons are at the bottom right.

Title

☒ Enable

Text
Max length is 25 characters

Text Color

Description

☒ Enable

Text
Max length is 125 characters

Text Color

Cancel | **Save**

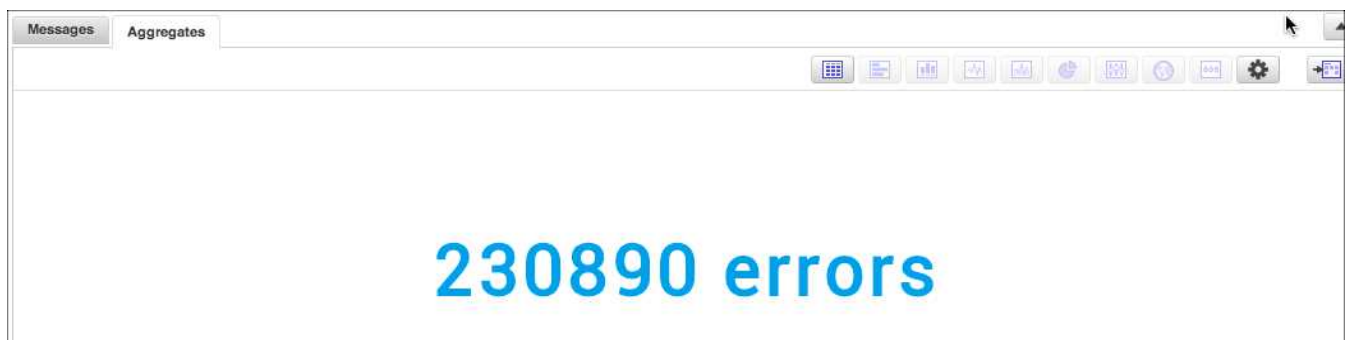
String Single Value Chart

A String Single Value chart displays the first result of a search query as a string, such as a top status or message.

For example, you can use the following query:

```
error | count | format("%d errors", _count) as errorString | fields errorString
```

which provides results in a Single Value chart like:



To create a String Single Value chart:

1. Run a query.
2. In the **Aggregate** tab, the chart is automatically displayed as a Single Value chart.

3. (Optional) Click **Add to Dashboard** if you'd like to save the chart as a Monitor.

To modify a String Single Value chart:

1. Select the **Properties** icon.
2. Select **Change Properties**. Here you can change the **Color** and **String Value**. Click **Save** to save changes.



The dialog box is titled "Change Viewer Properties". It has two main sections: "Color" and "String Value". In the "Color" section, there is a label "Override default color to" followed by a color selection box with a dropdown arrow. In the "String Value" section, there is a label "Use the following Description when data is unavailable:" followed by a text input field, the text "in the color", and another color selection box with a dropdown arrow. At the bottom right, there are "Cancel" and "Save" buttons.

Boolean Single Value Chart

A Boolean Single Value chart displays a value as true or false, such as when a system is up or down.

To create a Boolean Single Value chart, use a query such as:

*** | count as MyCount | if (MyCount>100,true,false) as MyCount**

which would produce results like:

MessagesAggregates

Page: 1 of 1

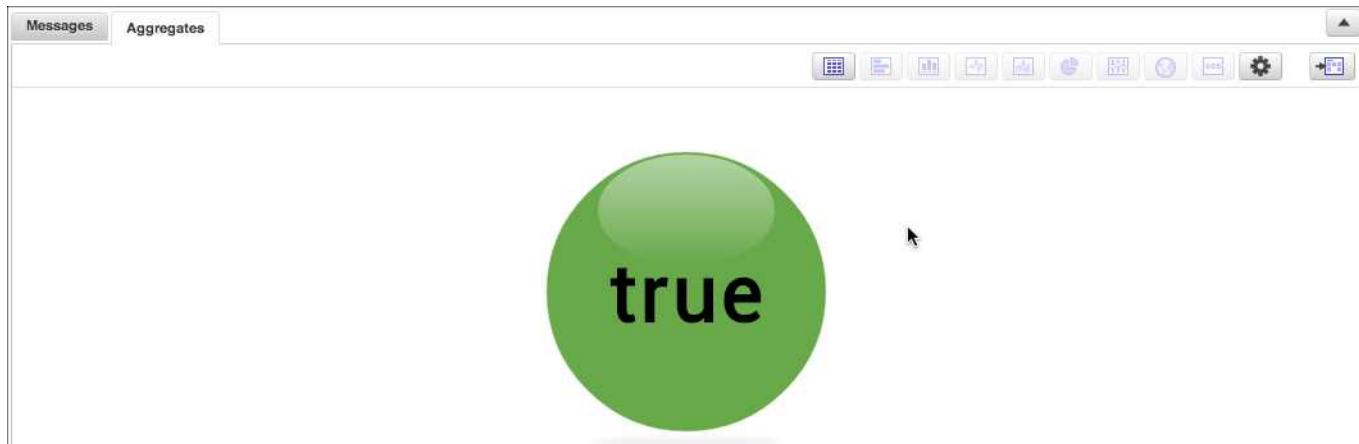
#	mycount
1	true

To create a Boolean Single Value chart:

1. Run a Boolean query.
2. In the **Aggregate** tab, choose the **Single Value** icon.



This displays the the result graphically.




3. (Optional) Click **Add to Dashboard** if you'd like to save the chart as a Monitor.

To modify a Boolean Single Value chart:

1. Select the **Properties** icon.
2. Select **Change Properties**. Properties are automatically configured for a Boolean Single Value chart to display as green for true or red for false, but you can customize the color values for your chart. Click **Save** to save any changes.

A screenshot of a dialog box titled 'Change Viewer Properties'. The dialog box has a close button (X) in the top right corner. It contains two main sections: 'Color' and 'Boolean Value'. Under the 'Color' section, there is a label 'Override default color to' followed by a dropdown menu showing a white square. Under the 'Boolean Value' section, there are two labels: 'Color for true value' followed by a dropdown menu showing a green square, and 'Color for false value' followed by a dropdown menu showing a red square. At the bottom of the dialog box, there are two buttons: 'Cancel' and 'Save'.

3. **Manage Labels.** This option allows you to enable a **Title** and a **Description**, including the text displayed and the color of the text. Click **Save** to save changes.



The image shows a 'Manage Labels' dialog box with a title bar and a close button. It contains two sections: 'Title' and 'Description'. Each section has an 'Enable' checkbox, a 'Text' input field, a character limit note, and a 'Text Color' color picker. The 'Title' section has a max length of 25 characters, and the 'Description' section has a max length of 125 characters. At the bottom right are 'Cancel' and 'Save' buttons.

Manage Labels

Title

☒ Enable

Text: Title text
Max length is 25 characters

Text Color: [Color Picker]

Description

☒ Enable

Text: Description text
Max length is 125 characters

Text Color: [Color Picker]

Cancel | Save

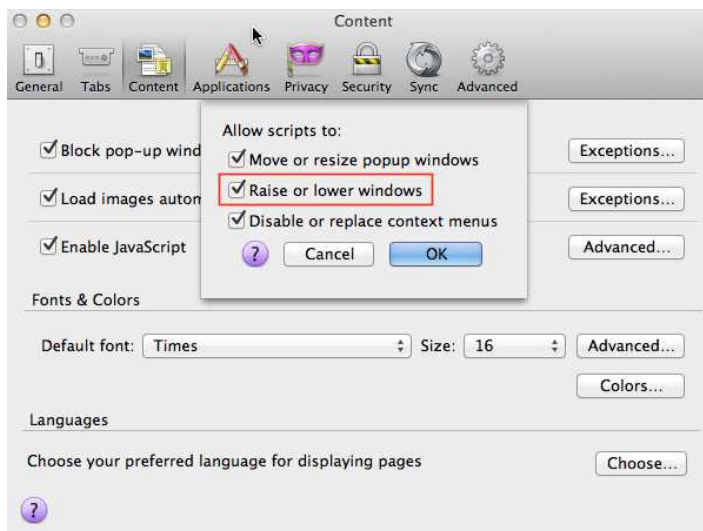
4. **Edit Settings JSON.** Allows you to directly edit the JSON code to change the appearance of the Single Value chart. Click **Save** to save changes.

Setting Firefox Permissions to View Dashboards

When using the Firefox browser to view Dashboards, you'll need to enable an option that allows Sumo Logic to change the order of open browser windows. In Firefox, it's referred to as raising and lowering (bringing to front or moving to back) windows.

To enable scripts to re-order windows in Firefox:

1. Log in to your Sumo Logic account on Firefox.
2. Do one of the following:
 - On PC, click **Options > Content > Enable JavaScript** and then click **Advanced**.
 - On Mac, choose **Preferences > Content > Enable JavaScript** and then click **Advanced**.
3. Select the **Raise or lower windows** option. Then click **OK**.



Using the Library

The Library provides a central location for shared and saved content in your Sumo Logic account, as well as content shared by others in your organization. In addition to shared and saved searches, Dashboards can be saved and shared in the Content Library.

As well as being a great tool for managing content, the Library allows you to launch searches and Dashboards with a single click—speeding up access to the searches you find yourself running consistently. Additionally, you can use the content that others in your organization have already developed to continually discover new insights in your data.

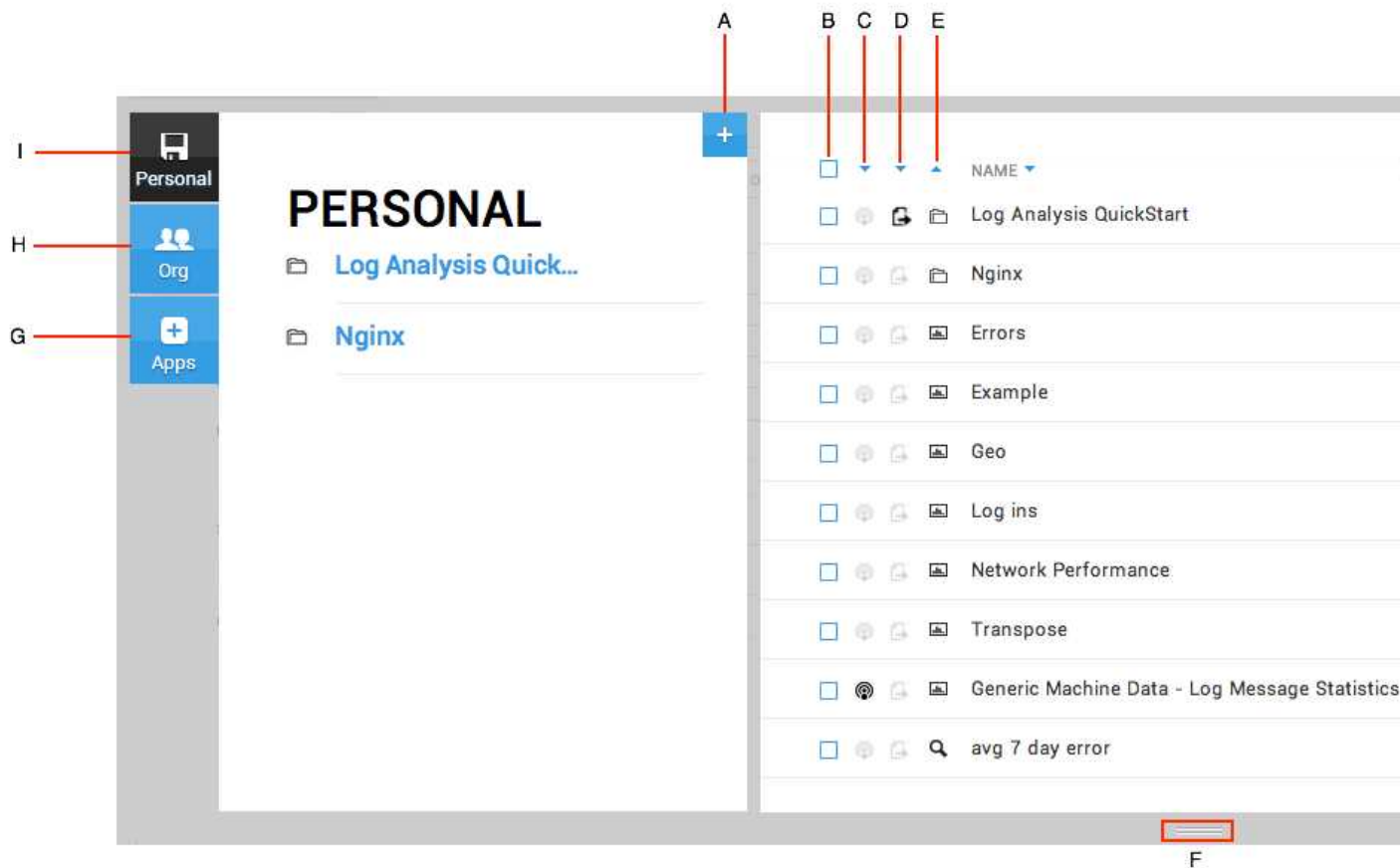


Any searches that you saved previously, as well as any Dashboards you've already built, will automatically appear in the Library.

How the Library works

The Library provides a single interface to organize and share searches and Dashboards, and install apps. By default, each user in an account has two folders: Personal and Org. Content stored in the **Personal** folder is private until a user publishes a search or Dashboard. Content in the **Org** folder is shared with all users across an organization's account.

Dashboards and searches are marked with icons to help you discern which content type you're viewing. You can launch a search or Dashboard by just clicking it.



- A. Creates a new folder.
- B. Click to select all.
- C. Click to sort by subscribed or unsubscribed status.
- D. Click to sort by published or unpublished status.
- E. Click to sort by Dashboard or search.
- F. Click to close the Library.
- G. Apps tab.
- H. Org tab. Displays content published by others in your organization.
- I. Personal tab. Displays your content.

What's the difference between sharing and publishing?

Any search or Dashboard you save to your **Org** folder is immediately made available to all users in your organization, exactly in the state the search or Dashboard in at the moment you shared it. A shared search is static—changes made to the search are not updated to other users. Shared content can be accessed by anyone in the organization. Think of shared content as being “pulled” by a user from a central repo.

Published content is slightly different. Once you publish a search or Dashboard, a user can **Subscribe** to the content, meaning that as you make changes to it, those changes are reflected in the user’s content as well. Think of published content as being “pushed” to a user who subscribes. Any changes are “pushed” (updated) to subscribers.

Searches and Dashboards can be published and unpublished at any time. Additionally, other users can copy what you’ve published and make further customizations.

Saving a Search

Whether you are running ad hoc searches during a forensic investigation or running standard searches for health checks, you can save any search to run later. When you save a search, you have the option to set up the saved search to run at a scheduled interval with an automated notification by email of the search results. You can edit a saved search at any time.

To save a search:

1. In the Search tab, after typing your search query, click **Save As** below the search field.



2. For **Search name**, enter a name for your Saved Search. If you'd like, type an optional description to help you identify this search. (Optional.)
3. The search query populates automatically in the **Search** field. You can make changes to the search syntax or query details if you need to.
4. Choose a **Time Range** option that will be the default range when you run the saved search. If you'd like to search backwards, you can type a time range like -15m.
5. Choose a **Folder** to save your search. To add a new Folder to the Library, click the blue "+" and name the new folder.

Scheduling searches

There are three ways to set up scheduled searches:

Scheduled email. The first option is to set up a scheduled search email, which provides a summary of the search results every time a scheduled search is run. The subject of this email is **Search Results**. No matter what the results of the search, you'll receive an email.

Alert email. Set up Alert Conditions if you'd rather just receive an alert email when certain conditions are met. This means you'd get an automated alert when or if parameters you set are triggered, based on the results of the scheduled search. For example, you could set up an alert if a certain number of users visit the free trial URL of your website. You can [run a search from this email](#) when you receive it.

You can be very specific with the alert condition—you can even set an exact number of results that triggers the email. Results can either be the number of log messages *OR* the number of aggregates returned by the saved search. If your saved search returns log messages, then the alert will use the number of messages you specify. If your query produces aggregate results, the alert will use the number of aggregates (or groups).

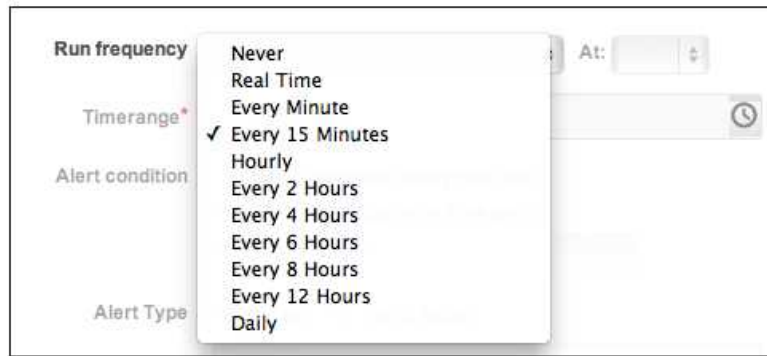
Real Time Alerts. Available only to Enterprise and paid trial customers, Real Time Alerts continuously monitor your Sumo Logic deployment, and return alert emails whenever conditions are met. You can learn more in [Setting up Real Time Alerts](#).



Scheduled searches are run according to the time zone of an individual's computer and browser, not according to the time zone of logs.

To schedule a search:

1. Do one of the following:
 - Open a saved search and convert it to a scheduled search. Click the **Open** link below the Search query field. Select the search you want to schedule, and then click **Edit**.
 - Use your current search query and then set up a schedule for it to periodically run and send alerts by email.
2. Choose an option from the **Run Frequency** menu:



- **Never.** Choose this option to temporarily turn off a scheduled search.
- **Real Time.** Enterprise and paid trial customers can use this option to set up **Real Time Alerts**.
- **Every Minute.** The search will run for the first time when you save the schedule, and then every minute after that.
- **Every 15 Minutes.** The search will run for the first time when you save the schedule, and then every 15

minutes after that.

- **Hourly.** The search will run for the first time at the top of the next hour after you save the schedule, and then every hour after that.
 - **Every 2, 4, 6, 8, or 12 Hours.** The search will run for the first time at the top of the hour you choose.
 - **Daily.** Choose the time you'd like to run the search every day. A Daily search will cover exactly 24 hours of activity. You can change the schedule whenever you'd like.
4. Choose a **Time Range** option to set the default range the scheduled search is run against. Alternately type a time range; for example, -15m to run the search against data generated in the past 15 minutes.
 5. For **Alert Condition**, choose one of the following:
 - **Notify me every time upon search completion** if you want an email with search results every time the search is run (depending on the frequency, you could get an email every 15 minutes, every hour, or once a day).
 - **Notify me only if the condition below is satisfied** if you'd like to set up a scheduled search that alerts you to specific events, and then set any of the following conditions before typing a value in the text box:

Number of Results. Depending on the search, set a condition to receive an email by the number of results. If your saved search returns log messages, then the alert will use the number messages you specify; if your query produces aggregate results, the alert will use the number of aggregates (or groups).

Equal to. Choose if there is an exact number of records in a search result at which you want to be notified.

Greater than. Choose if you want to be notified only if the search results include greater than that number of messages or groups you set in the text box.

7. For **Alert Type**, choose **Email** to receive email alerts.
8. For **Recipients**, enter one or more email addresses separated by commas for the recipients of the search results. You can edit the recipients at any time.
9. Click **Save**.

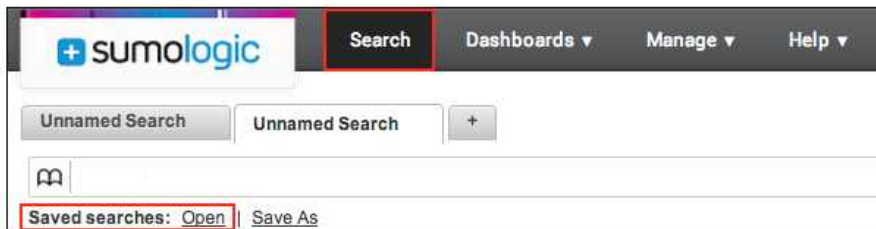
An email notification will be sent to the recipient email address(es) based on the notification parameters. For queries returning high numbers of messages or group results, the email notification contains a representative sample of the first 20 messages or groups.

Canceling or Editing Scheduled Searches

You can edit or cancel a scheduled search at any time. If you cancel a scheduled search, the search reverts to a saved search.

To cancel a scheduled search:

1. Click the **Open** link below the Search field.



2. Select the search you want to cancel, and then click **Edit**.
3. Choose **Never** from the **Run Frequency** menu to cancel the scheduled search, then click **Save**.

To edit a scheduled search:

1. Click the **Open** link below the Search query field.

2. Click **Edit** next to the scheduled search you'd like to change.
3. Make the changes, and then click **Save**.

Running a Search from an Alert Email

When you receive an alert email from a scheduled search, only a representative sample of the results are included. (We don't want to flood your email with hundreds of thousands of search results.)

Do either of the following:

- To see the complete results of the search, click the **View results in Sumo Logic** link in the email. Sumo Logic will recreate the search exactly matching the query and time parameters of the original scheduled search.

- To make changes to the search query before you run it again, click the search name (next to **Saved Search**) to open the Sumo Logic Web Application with the query entered in the search field.

Search Alert: Less than 550 results found for "Auth.log at 22h - 23h" Inbox x

Sumo Logic service@sumologic.com
to dev ▾

Saved Search	Auth.log at 22h - 23h
Search String	_sourcecategory=auth_log
Time Range	06/22/2012 17:00:00.000 to 06/22/2012 18:00:00.000
Run At	06/21/2012 19:01:11.770 for Prod Demo Data Feeder

Message Distribution [\(View results in Sumo Logic\)](#)

Most Recent Results

#	Time	Message
1	06/22/2012 17:59:58.000	Jun 22 2012 17:59:58 mahler sshd[24151]: Failed password for invalid user oracle from 10.8.200.114 port 32902 ssh2 Host: linux_box Name: /var/log/auth.log.2012-06-22 Category: auth_log
2	06/22/2012 17:59:54.000	Jun 22 2012 17:59:54 mahler sshd[24151]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.8.200.114 Host: linux_box Name: /var/log/auth.log.2012-06-22 Category: auth_log
3	06/22/2012 17:59:45.000	Jun 22 2012 17:59:45 mahler sshd[24151]: pam_unix(sshd:auth): check pass; user unknown Host: linux_box Name: /var/log/auth.log.2012-06-22 Category: auth_log
4	06/22/2012 17:59:42.000	Jun 22 2012 17:59:42 mahler sshd[24151]: Invalid user oracle from 10.8.200.114 Host: linux_box Name: /var/log/auth.log.2012-06-22 Category: auth_log
5	06/22/2012 17:59:37.000	Jun 22 2012 17:59:37 mahler sshd[24148]: Failed password for root from 10.8.200.114 port 44254 ssh2 Host: linux_box Name: /var/log/auth.log.2012-06-22 Category: auth_log
6	06/22/2012 17:59:36.000	Jun 22 2012 17:59:36 mahler sshd[24148]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.8.200.114 user=root Host: linux_box Name: /var/log/auth.log.2012-06-22 Category: auth_log
7	06/22/2012 17:59:34.000	Jun 22 2012 17:59:34 mahler CRON[24142]: pam_unix(cron:session): session closed for user root



If you are a new user and someone has forwarded an alert email, the links to the search will not work until you've completed your setup process.

Setting up Real Time Alerts



This feature is only available to customers with Enterprise accounts, or those in paid trials. If you'd like to try using Real Time Alerts, please contact your sales representative.

Real Time Alerts are scheduled searches that run nearly continuously. That means that you're informed in real time when error conditions exist.

When an alert condition is satisfied, Sumo Logic sends an email (or trigger a script action). Further emails are not sent while the condition remain satisfied. Instead, if the number of matches goes below the threshold, and then exceeds the threshold a second time we send a second alert.



If the time zone of messages is set incorrectly, those logs won't be picked up by Real Time Alerts.

Operator Limitations for Real Time Alerts

Some queries cannot be used in Real Time Alert searches. Other operators can be used in Real Time search, but in the search, they must be included after the first "group-by" phrase:

Not supported for Real Time Alerts	Must be added after a "group by" phrase
<ul style="list-style-type: none">• Count_frequent• Details• First• Last• Join• Parse• Save• Sessionize• Summarize• Trace	<ul style="list-style-type: none">• Accum• Diff• Smooth• Sort• Top• Total

Set Up Real Time Alerts

To set up Real Time alerting:

1. In the Library, highlight a search, then select **Edit**. You can also use the query currently in the search text box; just click Save As under the box.



2. Choose **Real Time** from the **Run Frequency** menu.
3. Choose an option from the **Time Range** menu.
4. For **Alert Condition**, set the Number of Results conditions:
 - **Equal to**. Choose if there is an exact number of records in a search result at which you want to be notified. Type that number in the text box.

- **Greater than.** Choose if you want to be notified only if the search results are greater than a number of messages or groups. Type that number in the text box.
 - **Greater than or equal to.** Choose if you want to be notified only if the search results are greater than or equal to a number of messages or groups. Type that number in the text box.
 - **Fewer than.** Choose if you want to be notified only if the search results are less than a number of messages or groups. Type that number in the text box.
 - **Fewer than or equal to.** Choose if you want to be notified only if the search results are less than or equal to a number of messages or groups. Type that number in the text box.
5. For **Alert Type**, choose **Email** to receive email alerts, or choose Script Action to return search results to a Script Action that you've configured. See [Collecting from a Script Action](#).
 6. If you've chosen to be alerted by email, for **Recipients**, enter email addresses for the recipients of the alert results. You can edit the recipients at any time.
 7. Click **Save**.

Save Search

Search name*
Failed logins

Description

Search*

```
( "failed login for user" )
| parse "for user '*'" as user
| count by user
```

Timerange
Last 15 Minutes

Folder
PERSONAL

Run frequency
Real Time

Timerange*
Last 3 Minutes

Please choose a time range of 20 minutes or less.

Alert condition
☐ Send notification every time upon search completion
☒ Send notification only if the condition below is satisfied:

Number of results
Greater than or equal to >= 1

Alert Type
Email

Recipients*
rosemary@sumologic.com

Separate email addresses with commas.

Cancel Save

An email notification will be sent to the recipient email address(es) based on the notification parameters. For queries returning high numbers of messages or group results, the email notification contains a representative sample of the first 20 messages or groups.

Publishing a search from the Library

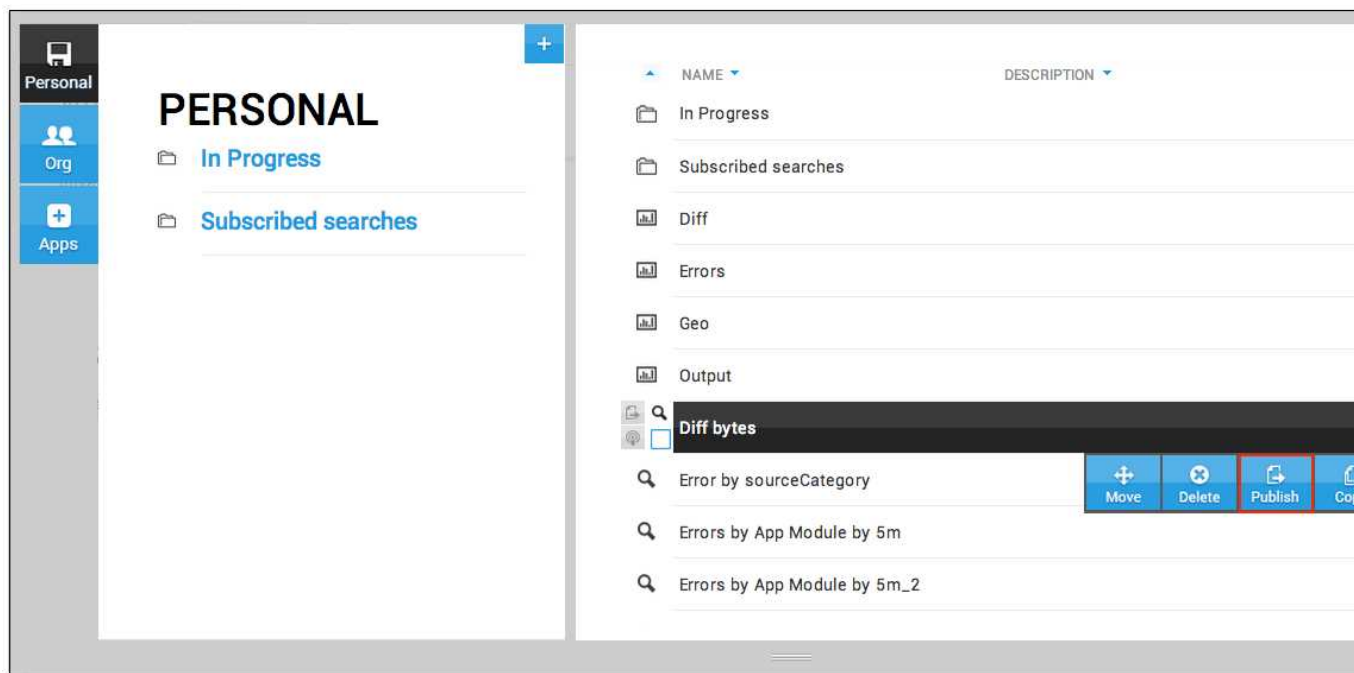
Once you've saved content, you can choose to **Publish** it, making it available for subscription. Published content is automatically added to your **Org** folder.

A Published search is slightly different from a saved search. Once you publish a search, a user can **Subscribe** to it, meaning that as you make changes to it, those changes are reflected in the other user's Library as well. Think of a published search as being "pushed" to a user who subscribes; any changes are pushed (updated) to subscribers.

Searches and Dashboards can be published and unpublished at any time. Additionally, other users can copy what you've published and make further customizations.

To publish a search:

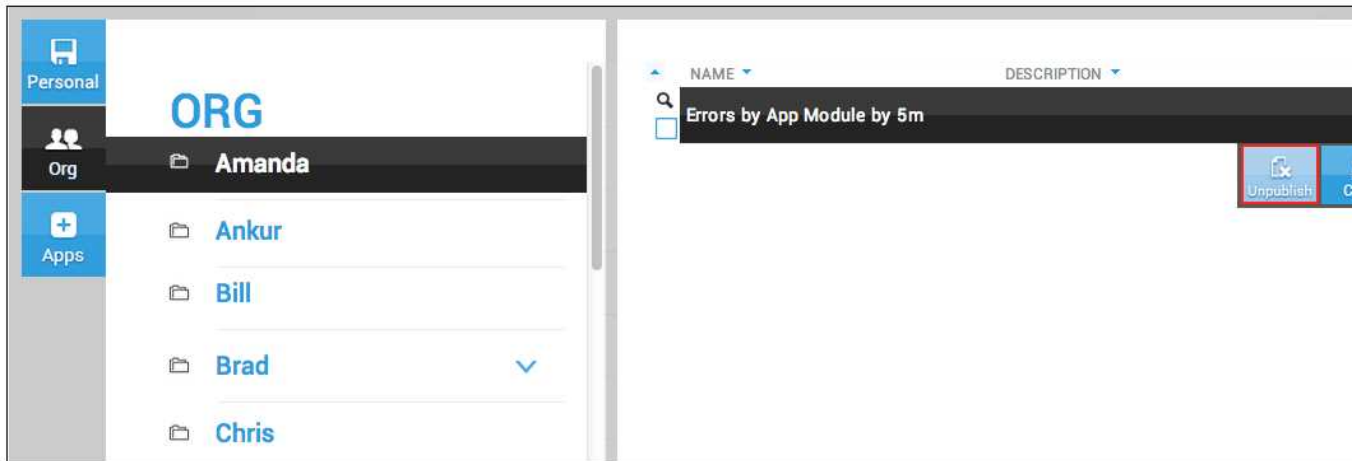
1. Click a search, then click the double arrow to the right of the name. Click **Publish**.



2. When prompted to confirm the change, click **Publish**. The search is moved to your Org folder:

To unpublish a search:

1. In your **Org** folder, select a published search or Dashboard, then click the double arrow to the right of the name. Click **Unpublish**.



2. When prompted, click **OK**.

The content is moved to your Personal folder.

Publishing Dashboards

Publishing a Dashboard is a great way to keep everyone on top of data that is important to your organization. After a Dashboard is published others can subscribe to it, which means that any changes you make will also be reflected in the subscribers' accounts.

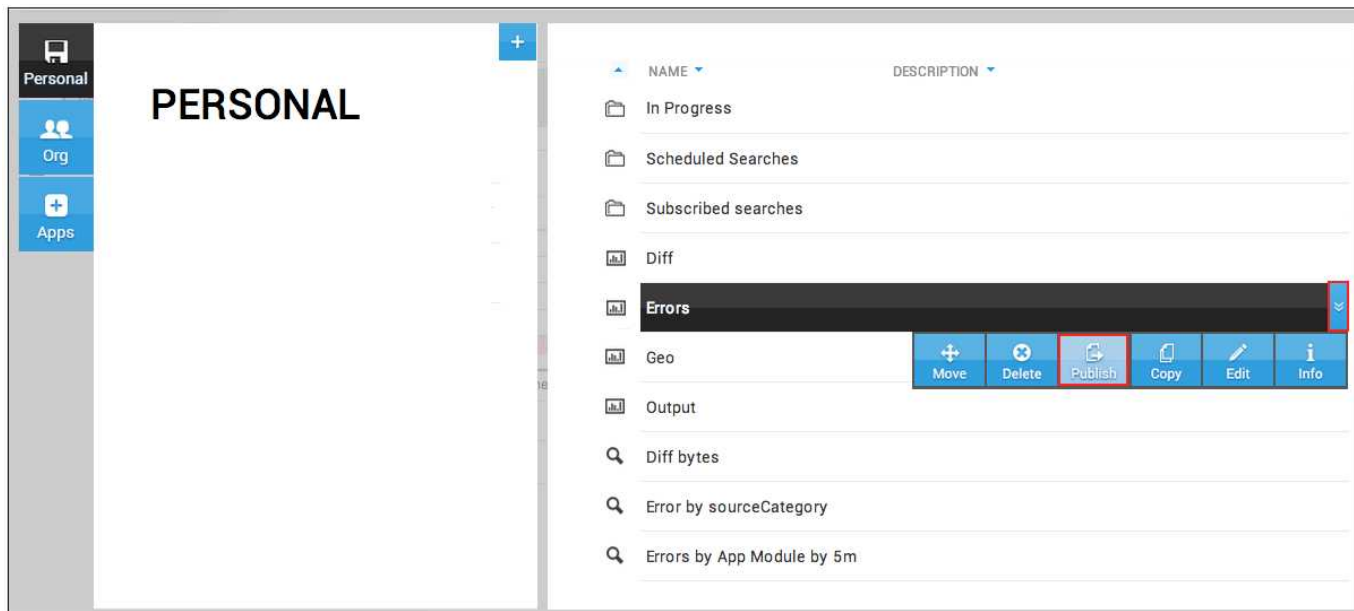
Once you publish a Dashboard, a user can **Subscribe** to it, meaning that as you make changes to it, those changes are reflected in the other user's Library as well. Think of a published Dashboard as being "pushed" to a user who subscribes; any changes are pushed (updated) to subscribers.



Others who view your published Dashboards can run queries based on Monitors you've saved in the Dashboard. Then they can save their own versions.

Publishing Dashboards from the Library

1. Click a Dashboard, then click the double arrow to the right of the name. Click **Publish**.



2. When prompted to confirm the change, click **Publish**. The search is moved to your Org folder:

To unpublish a Dashboard:

1. In your **Org** folder, select a published Dashboard, then click the double arrow to the right of the name. Click **Unpublish**.
2. When prompted, click **OK**.

The content is moved to your Personal folder.

Publishing from the Dashboards page

In addition to using the Library, you can choose to publish (or unpublish) a Dashboard directly from the Dashboard menu.

To publish a Dashboard:

1. Click the Dashboard's **Properties** icon, located right next to the Dashboard's name.
2. Choose **Publish Dashboard**.



To unpublish a Dashboard:

1. Click the Dashboard's **Properties** icon.
2. Choose **Unpublish Dashboard**.

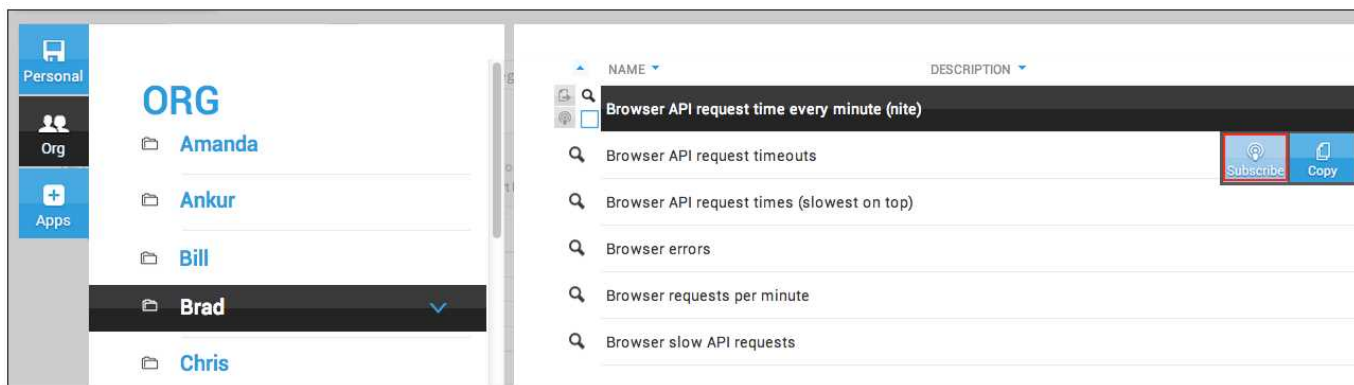


Subscribing to searches and Dashboards

You can subscribe to content stored in other user's Shared folder. Once you subscribe, you'll choose a folder to store the search or Dashboard; any changes the owner makes will be reflected in the copy stored in your library.

To subscribe:

1. In the Content Library, click **Org**. This opens the list of folders used by others in your organization.
2. Click the folder that contains the search or Dashboard, then click the search or Dashboard you'd like to subscribe to.
3. Click the double-arrow to the right, then click **Subscribe**.



4. When asked to confirm the subscription, make any changes to the name of the search or Dashboard, and then choose a location to save the content. Then click **Subscribe**.

Tip: If you'd like to create a new folder, click the blue "+" icon.

Copying Content

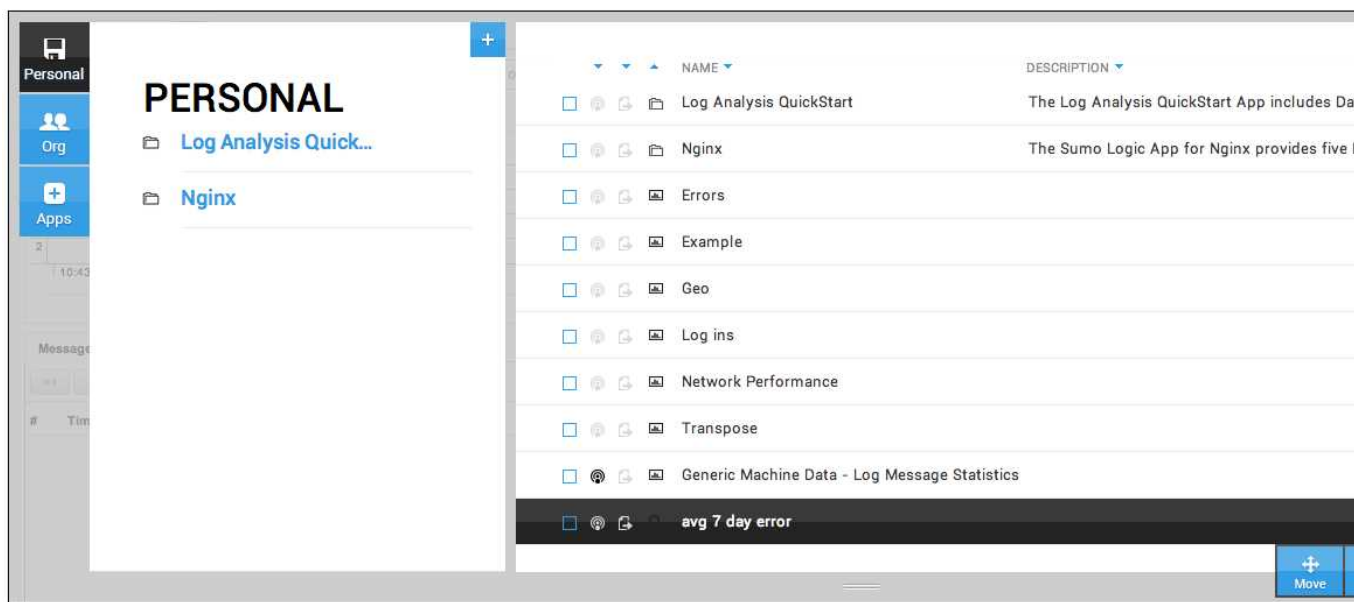
In addition to subscribing to a search or Dashboard, you can make a copy of an item. Once you make a copy, you can edit the search or Dashboard to make it fit your needs. Copied content is not kept up-to-date like content you've subscribed to.

You can also make a copy of a folder, which grabs all the content saved in the folder, then adds the copied folder to your Personal directory.

Any copied content is moved to your Personal folder; you can choose to save it in an existing subfolder or create a new subfolder.

To copy an item from your Personal folder:

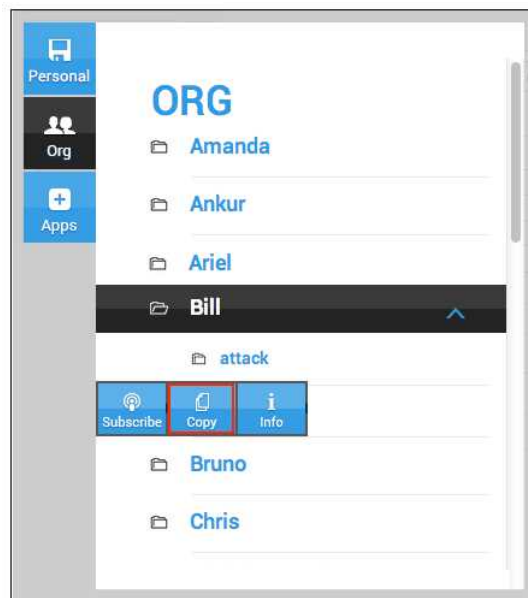
1. Select the item.
2. Click **Copy** in the menu.
3. Choose the location where you'd like to save the copied content and click **Copy**.



To copy an item from an Org-level folder:

1. Select the item, either a folder, a search, or a Dashboard.
2. Click **Copy** in the menu.

3. Choose the location where you'd like to save the copied content and click **Copy**.



Installing Apps from the Library

The **Library** makes it easy to install the Sumo Logic Applications your organization needs.

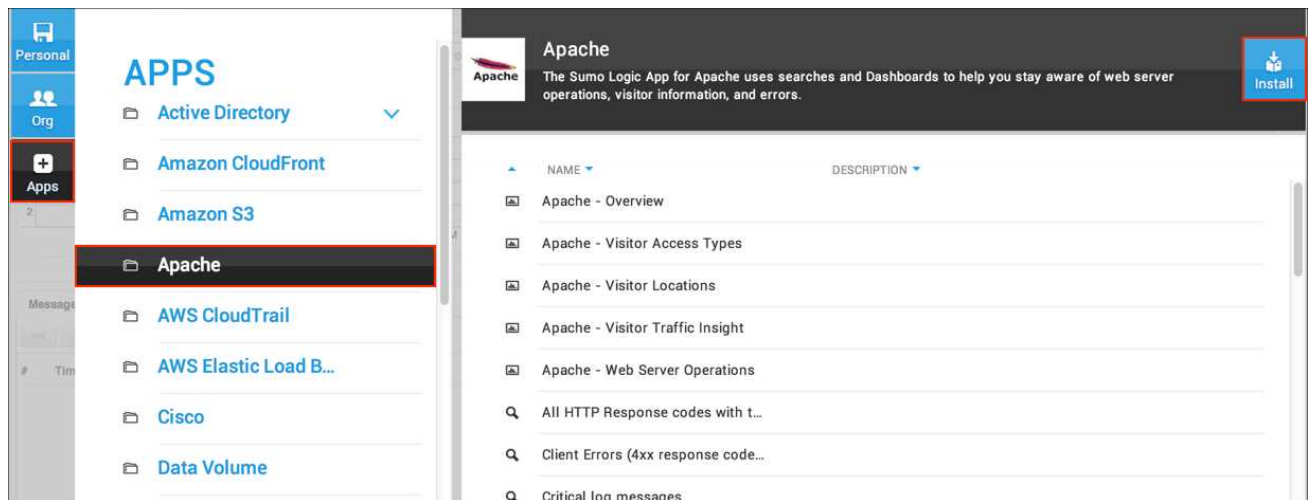
Certain apps have specific installation requirements. Be sure to check [Sumo Logic Apps](#) for more information.



Applications can be installed only by users with Admin permissions in a **Sumo Logic Professional** or a **Sumo Logic Enterprise** account. Organizations with a **Sumo Logic Free** account currently cannot use Sumo Logic Applications.

To install an Application:

1. In the **Library**, click the **Apps** tab.
2. Click the name of the app you'd like to install.
3. Click **Install**.

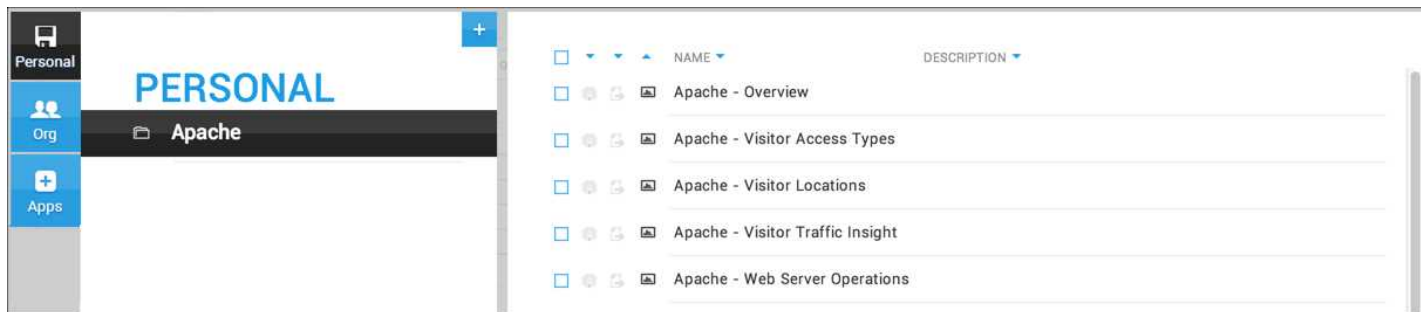


4. In the **Install Application** dialog box, do one of the following for **Data Source Mapping**:
 - **Select from _sourceCategory values.** Choose an existing _sourceCategory present in your account.
Important: If you do not select the correct _sourceCategory, data will not be loaded into the app.

- **Custom data filter.** To set up a specific data filter, type the keyword(s) you'd like to use to filter the data.

5. For **Folder**, choose either the **Personal** folder or a subfolder in the **Personal** folder. (Click the blue + to create a new subfolder).
6. Click **Install**.

After the installation is completed, you can see the App in your **Personal** folder:



After installing an app you'll need to start each Dashboard, as described in [Starting Dashboards](#).

Using Library keyboard shortcuts

The Library needs to be open in order to use keyboard shortcuts.

Shortcut	Action
I + m	Opens the actions menu for a folder. If a selected folder doesn't have a menu (such as the Personal folder) a message displays.
I + a	Selects all items in a selected folder.
I + o	Opens the active folder.
I + c	Closes the Library.
esc	Closes the Library

Sumo Logic Applications

Sumo Logic Applications deliver out-of-the box dashboards, reports, saved searches, and field extraction for popular data sources. When an app is installed in Sumo Logic, these pre-set searches and Dashboards are customized with your source configurations and populated in a folder selected by you.

Applications address five common use cases:

- **Increase availability and performance.** Sumo Logic enables issues to be identified before they impact the application and customer. Precise, proactive analytics quickly uncover hidden root causes across all layers of the application and infrastructure stack.
- **Provide real time insights.** With Sumo Logic enterprises easily extract machine data insights to provide greater intelligence around their customers, products, and application usage. These insights provide a more accurate and complete analysis for business users.
- **Accelerate Cloud deployment.** Sumo Logic enables enterprises to automate and speed the development and deployment process for cloud-based applications. Companies can rapidly detect, identify and resolve application issues.
- **Decrease app time to market.** With Sumo Logic, companies can implement a consistent release process resulting in on-time releases. They can easily identify application issues and configuration changes across development, test and deployment environments.
- **Enforce compliance.** Sumo Logic delivers a simple, proactive and automated process to audit and investigate operational, security and regulatory compliance incidents. All data is centralized, secured, and easily analyzed in real time through a single, highly scalable solution.

Sumo Logic Log Analysis QuickStart Application

The Log Analysis QuickStart Application, created especially for new users of Sumo Logic, includes searches to extract important information from your log files, independent of where they get generated. Whether you are new to log management or plan to migrate from other products, the Log Analysis QuickStart app will bring you up to speed with the Sumo Logic search, visualization, and analytics capabilities.

Ready to install the app? See [Installing the Log Analysis QuickStart App](#).

Sumo Logic Log Analysis QuickStart App Dashboards

The Log Analysis QuickStart Application, created especially for new users of Sumo Logic, includes searches to extract important information from your log files, independent of where they get generated. Whether you are new to log management or plan to migrate from other products, the Log Analysis QuickStart app will bring you up to speed with the Sumo Logic search, visualization, and analytics capabilities.

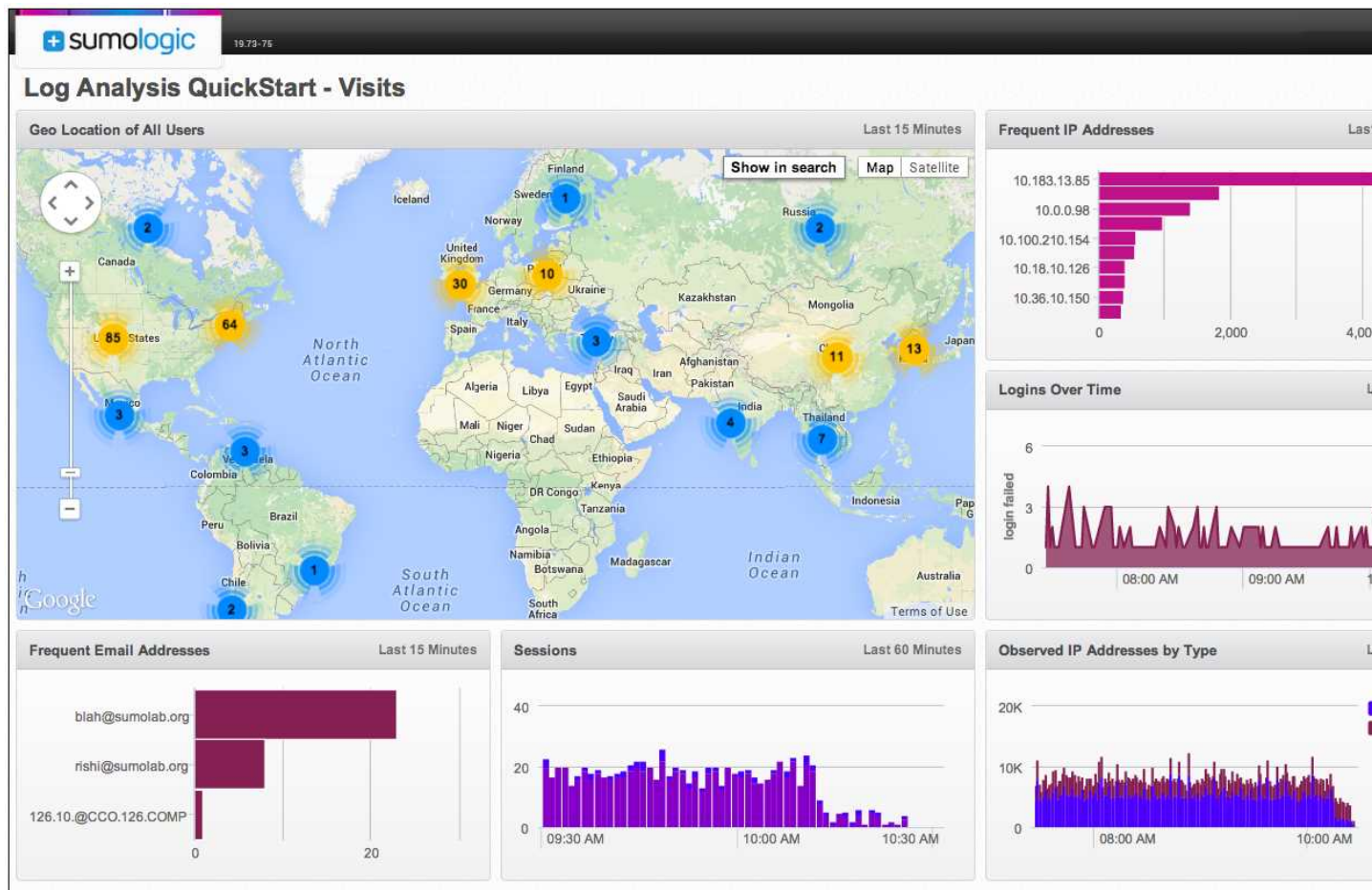
Ready to install the app? See [Installing the Log Analysis QuickStart App](#).

Dashboards

There are three Dashboards that give you an easy way to learn how Sumo Logic can help you monitor and analyze events occurring from your overall deployment.

Visits Dashboard

The Visits Dashboard displays identifying information about external and internal visitors across your deployment, including email addresses visitors are using.



Geo Location of All Users. Uses a geo location search to display the locations of IP addresses used by visitors.

Frequent IP Addresses. Shows a list of the most frequently used IP addresses by visitors.

Logins Over Time. Displays the successful and failed logins over the past three hours.

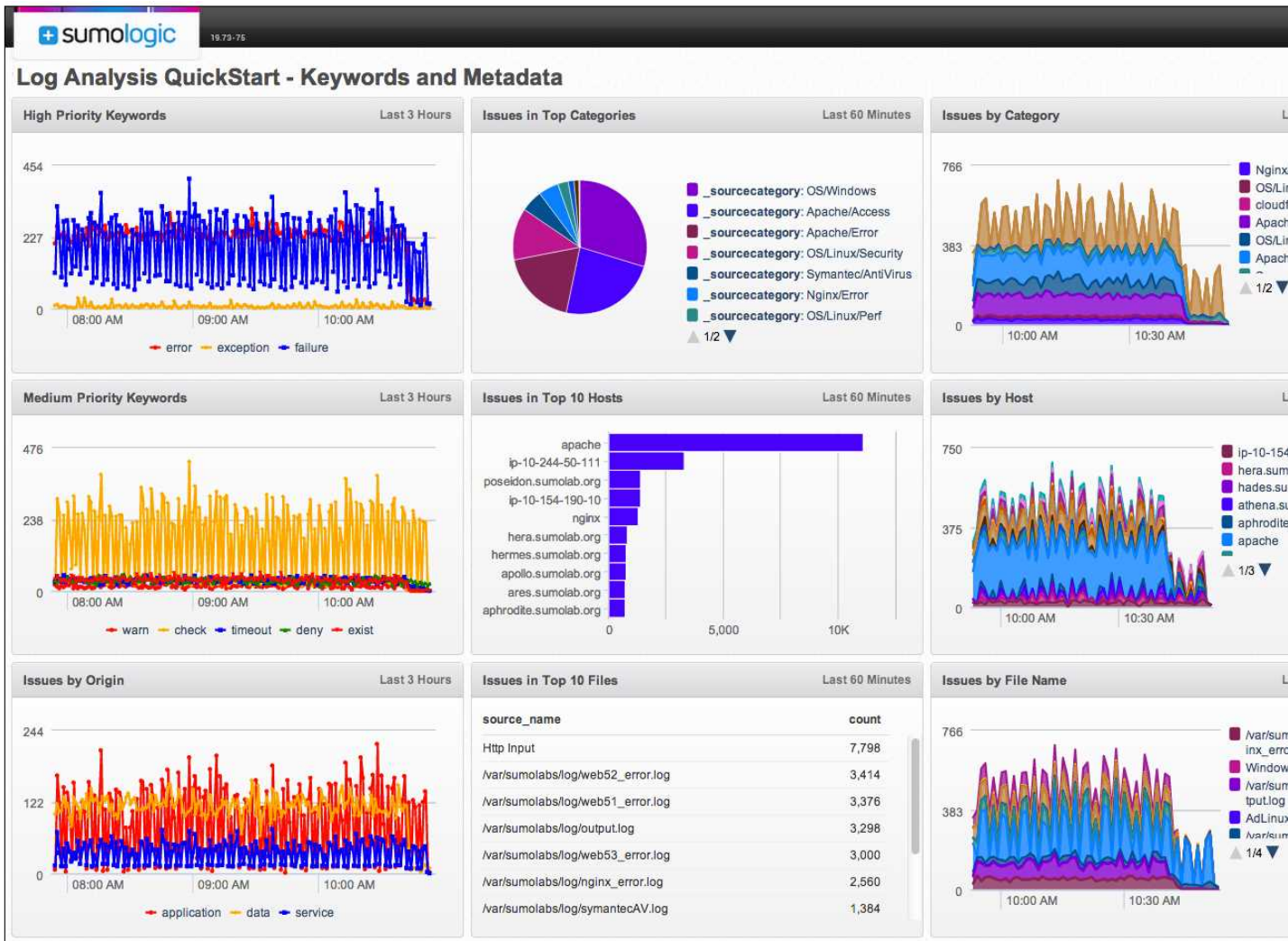
Observed IP Addresses by Type. Displays IP addresses used by internal and external visitors.

Sessions. Monitors errors across all sessions in your deployment.

Frequent Email Addresses. Displays the most frequently used email addresses.

Keywords and Metadata Dashboard

This Dashboard provides several ways to monitor your logs based on the metadata of your data. All of the Monitors include searches for specific issues. You can learn more about metadata options in [Establishing Metadata Conventions](#).



High Priority Keywords. Displays how often the terms error, failure, and exception appear in all log messages over time.

Medium Priority Keywords. Displays how often the terms time out, warn, check, exist, reject, deny, and timeout appear in all log messages over time.

Issues by Origin. Displays how often high priority keywords occur by the origin of issue, which could be application, data or service.

Issues in Top Categories. This monitor shows the top 10 source categories by number of log messages that contain error, exception, or failure terms.

Issues in Top 10 Hosts. Displays the top 10 hosts by number of log messages that contain error, exception, or failure terms.

Issues in Top 10 Files. Shows the top 10 files by number of log messages that contain error, exception, or failure terms.

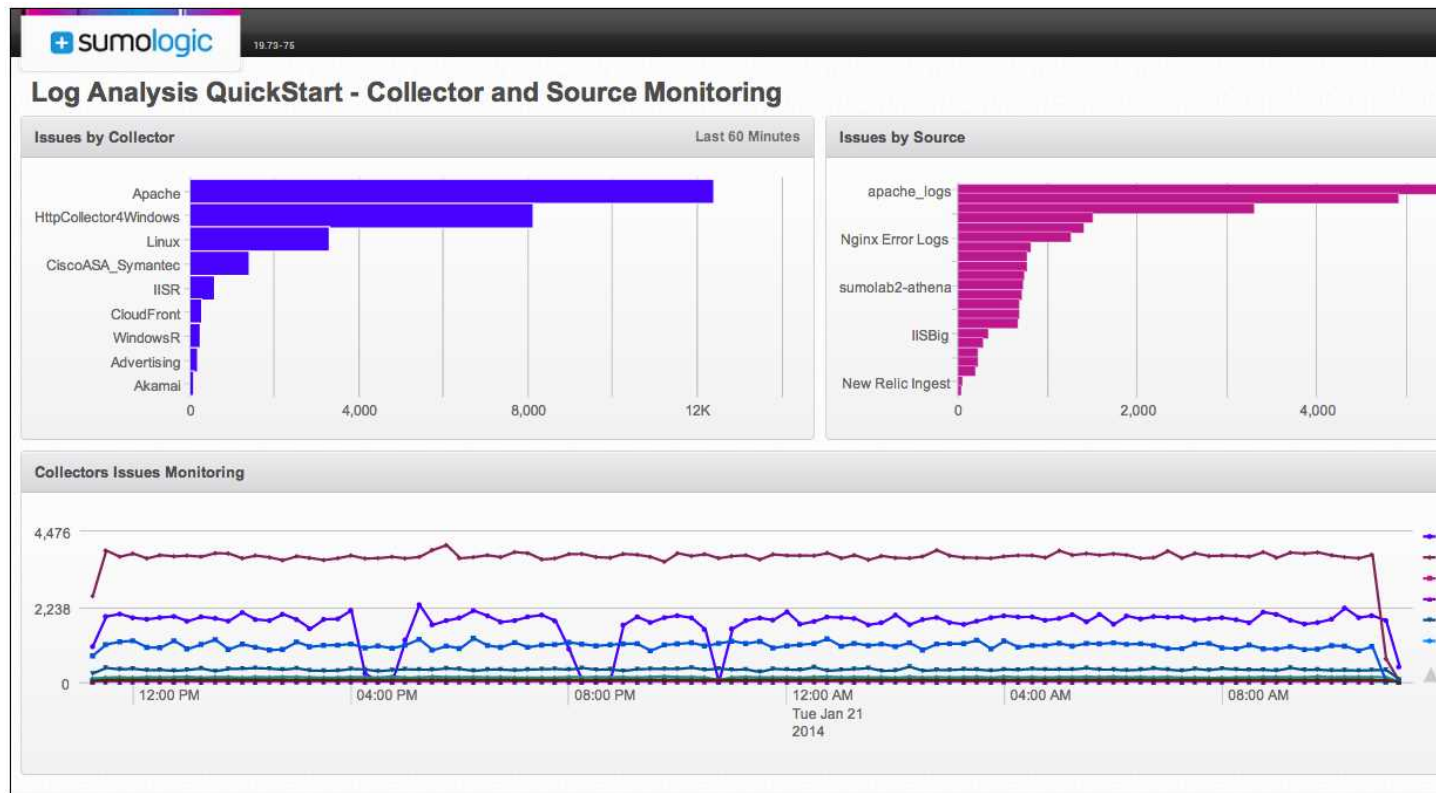
Issues by Category. This Monitor shows the number of log messages that contain error, exception, or failure terms by Source Category over time.

Issues by Host. Displays the number of log messages that contain time out, error, exceptions, and failures broken down by Source Host over time.

Issues by File Name. Displays the number of log messages that contain error, exception, or failure terms issues by log file name over time.

Collectors and Source Monitoring Dashboard

The Monitors in the Collector and Source Monitoring Dashboard help you keep an eye on the machines running Collectors and Sources. If a machine begins to have issues (such as no logs being uploaded to Sumo Logic) you'll know at a glance.



Issues by Collector. This Monitor displays the number of log messages that contain error, exception, or failure terms by Collector.

Issues by Source. Shows the number of log messages that contain error, exception, or failure terms by each Collectors' Source.

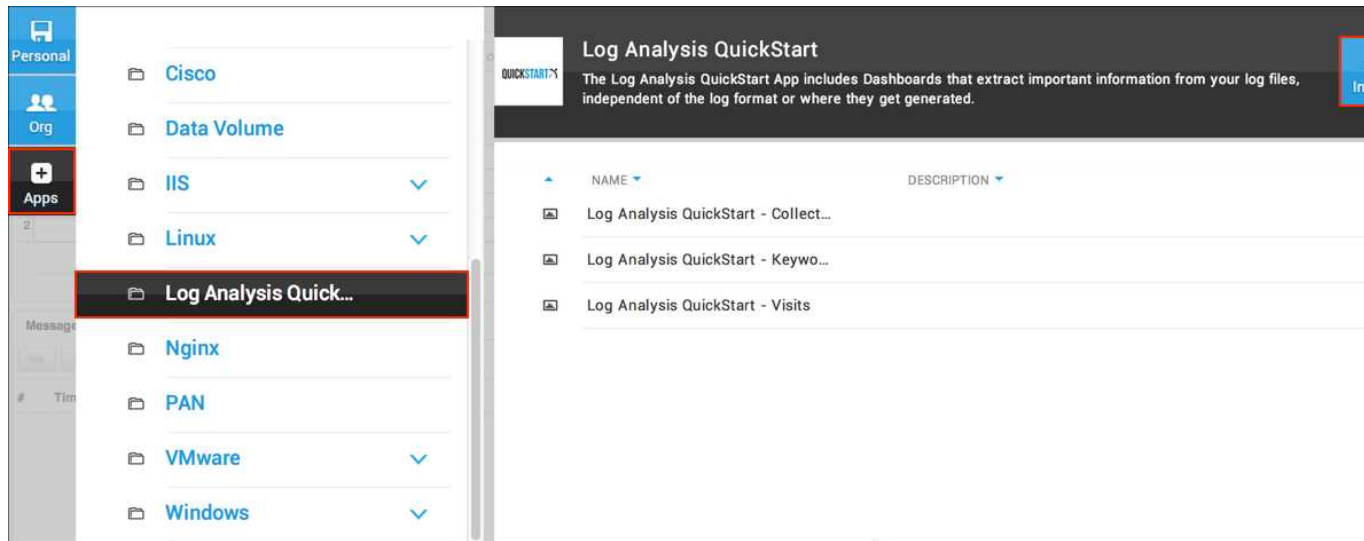
Collector Issue Monitoring. Displays warnings generated over time for each Collector in your deployment.

Installing and starting the Log Analysis QuickStart App

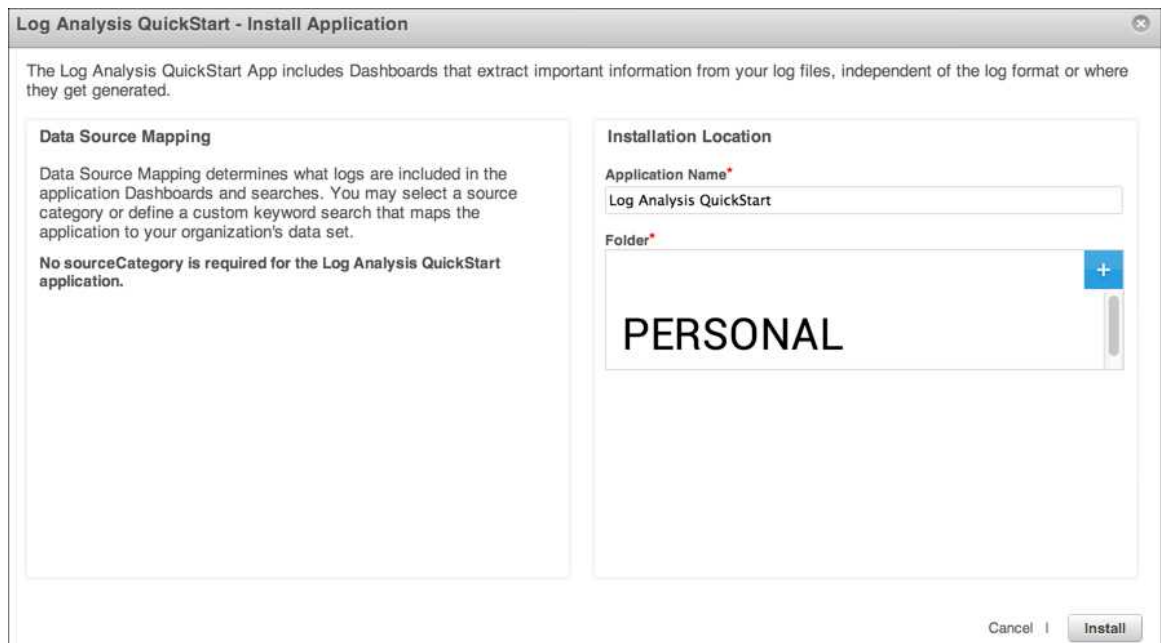
The Library feature of the Sumo LogicWeb Application allows an Admin to install the QuickStart App. Your organization will be up and running with the app in just a few minutes.

To install the app:

1. In the **Library**, click the **Apps** tab.
2. Click **Log Analysis QuickStart**.
3. Click **Install**.



4. In the **Install Application** dialog box, choose a location in the **Personal** folder.
5. Click **Install**.



Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

Sumo Logic App for Data Volume

The Sumo Logic app for Data Volume allows you to view at a glance your account's data usage volume by category, Collector, Source name, and hosts. The app uses predefined searches and a Dashboard that provide visibility into your environment for real time analysis of overall usage.

To use the Sumo Logic Data Volume app, an administrator must first enable the feature manually. For more information, see [Data Volume Index.htm](#).

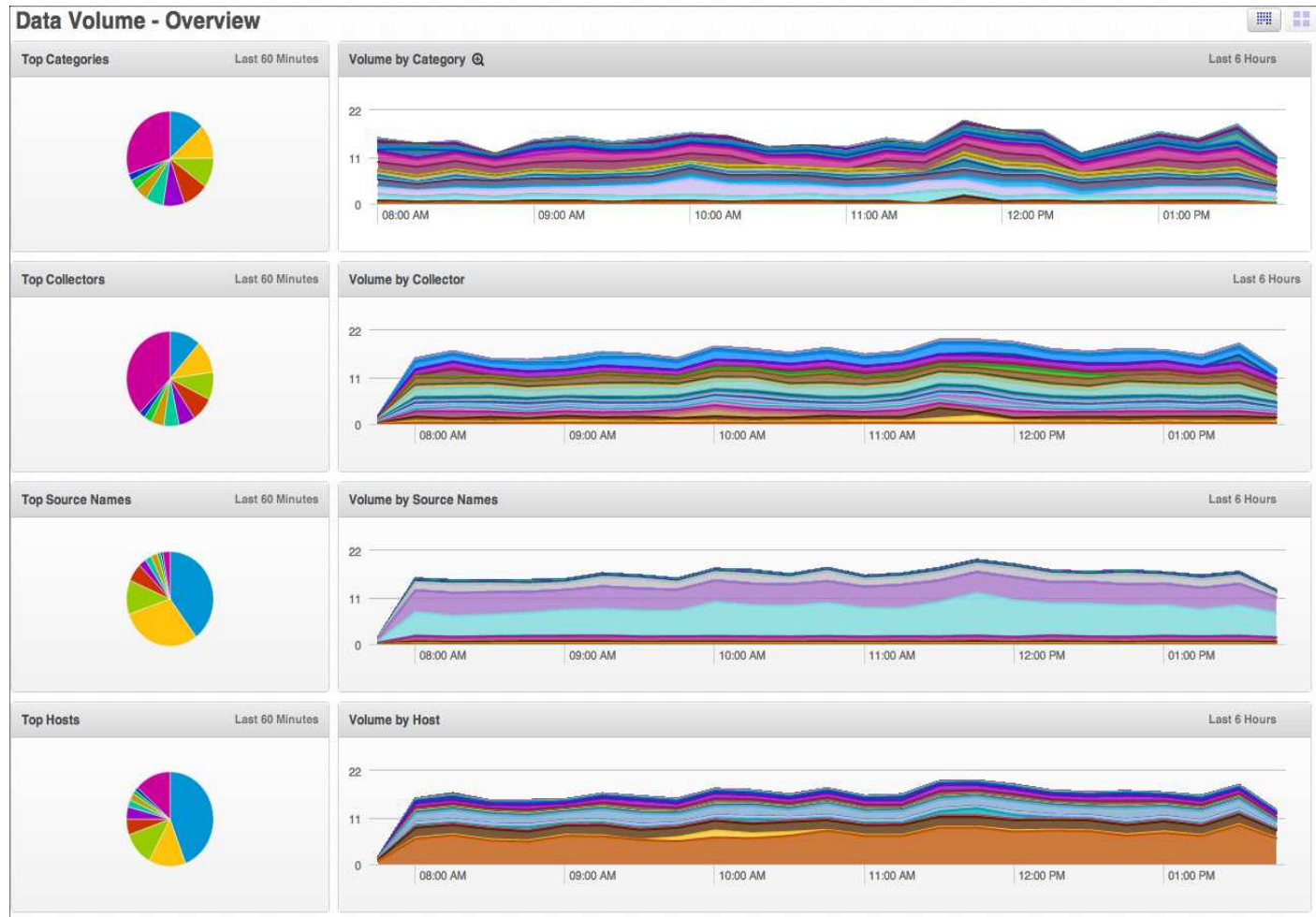
Once the Data Volume Index is enabled, volume data is not back filled to any time before the feature was enabled. Data is only provided from the time the feature is enabled forward.

Sumo Logic App for Data Volume Dashboards

For each Monitor in the Dashboard, you can perform the following actions:

- To display details on the Monitor time range, hover over the text in the top right corner.
- To zoom into the Monitor for more information, click the magnifying glass icon in the header.
- To view details in the **Search** page, click a section of the pie chart.

Data Volume - Overview



Top Categories. Displays the top 10 source categories and their volume usage in Bytes over the last 60 minutes, displayed in a pie chart.

Volume by Category. Shows your account's volume usage by category in GBytes for timeslices of 15 minutes over the last 6 hours, displayed in an area chart.

Top Collectors. Provides the top 10 Collectors and their volume usage in Bytes over the last 60 minutes, displayed in a pie chart.

Volume by Collector. Displays volume usage by Collector in GBytes for timeslices of 15 minutes over the last 6 hours, displayed in an area chart.

Top Source Names. Shows the top 10 Source names and their volume usage in Bytes over the last 60 minutes, displayed in a pie chart.

Volume by Source Name. Provides volume usage by Source name in GBytes for timeslices of 15 minutes over the last 6 hours, displayed in an area chart.

Top Hosts. Displays the top 10 hosts by volume usage in Bytes over the last 60 minutes, shown in a pie chart.

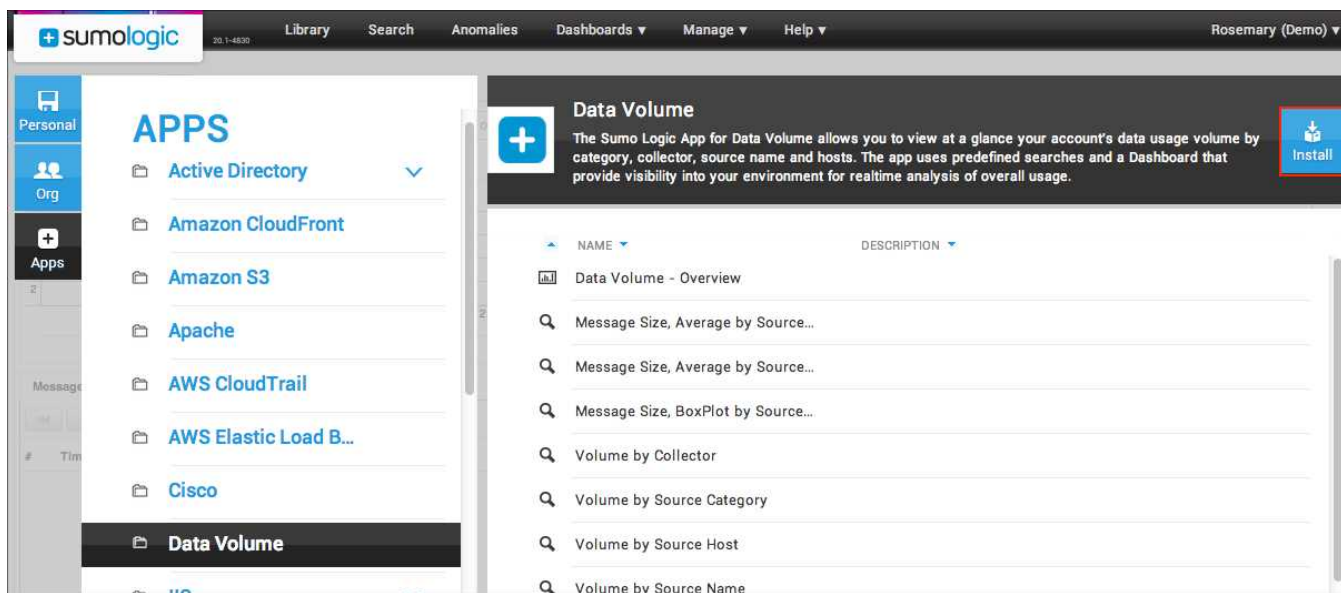
Volume by Host. Shows the volume usage of your account by host in GBytes for timeslices of 15 minutes over the last 6 hours, displayed in an area chart.

Installing the Data Volume App

The **Library** feature of the Sumo Logic Web Application allows an Admin to install the Data Volume App. Your organization will be up and running with the app in just a few minutes.

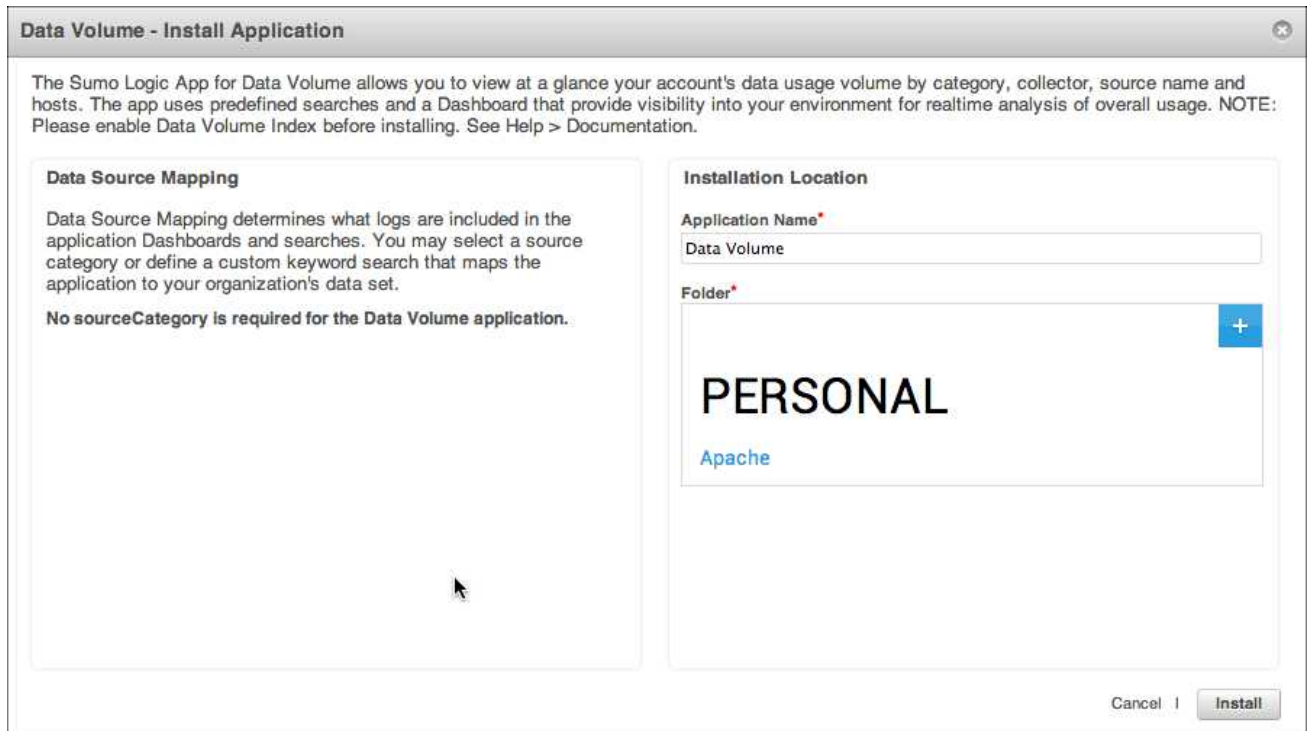
To install the app:

1. In the **Library**, click the **Apps** tab.
2. Click **Data Volume**.
3. Click **Install**.



4. In the **Install Application** dialog box, choose a location in the **Personal** folder.

5. Click Install.



The Sumo Logic App for Data Volume allows you to view at a glance your account's data usage volume by category, collector, source name and hosts. The app uses predefined searches and a Dashboard that provide visibility into your environment for realtime analysis of overall usage. NOTE: Please enable Data Volume Index before installing. See Help > Documentation.

Data Source Mapping

Data Source Mapping determines what logs are included in the application Dashboards and searches. You may select a source category or define a custom keyword search that maps the application to your organization's data set.

No sourceCategory is required for the Data Volume application.

Installation Location

Application Name*

Data Volume

Folder*

PERSONAL

Apache

Cancel | Install

Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

Sumo Logic App for Apache

The Sumo Logic Application for Apache gives insight into website visitor behavior patterns, monitors server operations, and assists in troubleshooting issues that span entire web server farms. The app consists of predefined parsers, searches, and Dashboards, providing visibility into your environment for real time or historical analysis.

Log Types

The Sumo Logic App for Apache assumes the NCSA extended/combined log file format for Access logs and the default Apache error log file format for error logs.

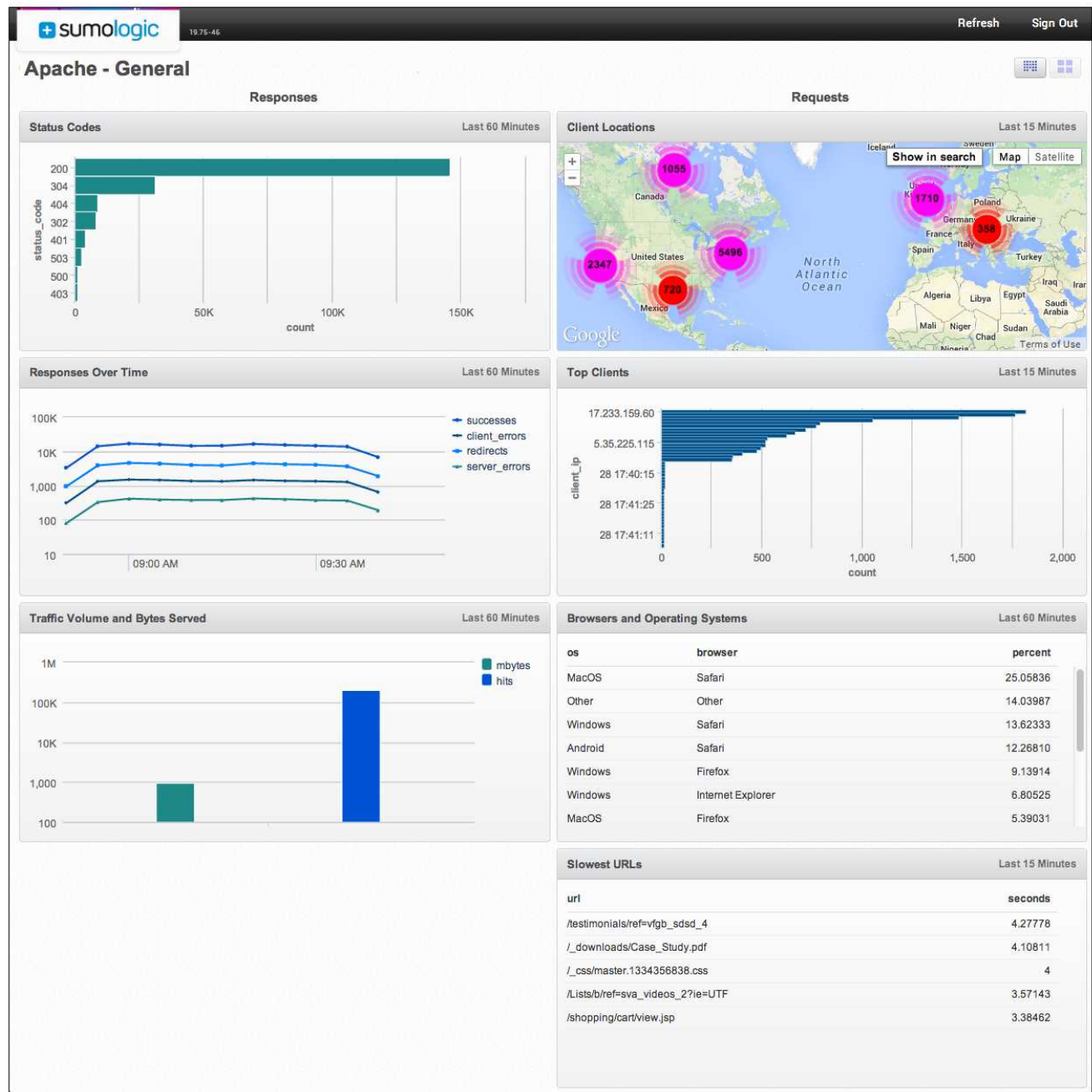
All Dashboards (except the Web Server Operations dashboard) assume the Access log format. The Web Server Operations Dashboard assumes both Access and Error log formats, so as to correlate information between the two.

For more details on the Access log file format, see http://httpd.apache.org/docs/current/mod/mod_log_config.html.

Sumo Logic App for Apache Dashboards

The Sumo Logic Application for Apache consists of predefined parsers, searches, and Dashboards, providing visibility into your environment for real time or historical analysis.

General Dashboard



Status Codes. Displays the number of 200, 202, 300, 304, 401, 404, 500, and 503 generated over the past hour, sorted by count.

Responses Over Time. Shows the successes, client errors, redirects, and server errors that occurred over the past hour.

Traffic Volume and Bytes Served. Megabytes served to visitors and the number of hits are shown in a column chart.

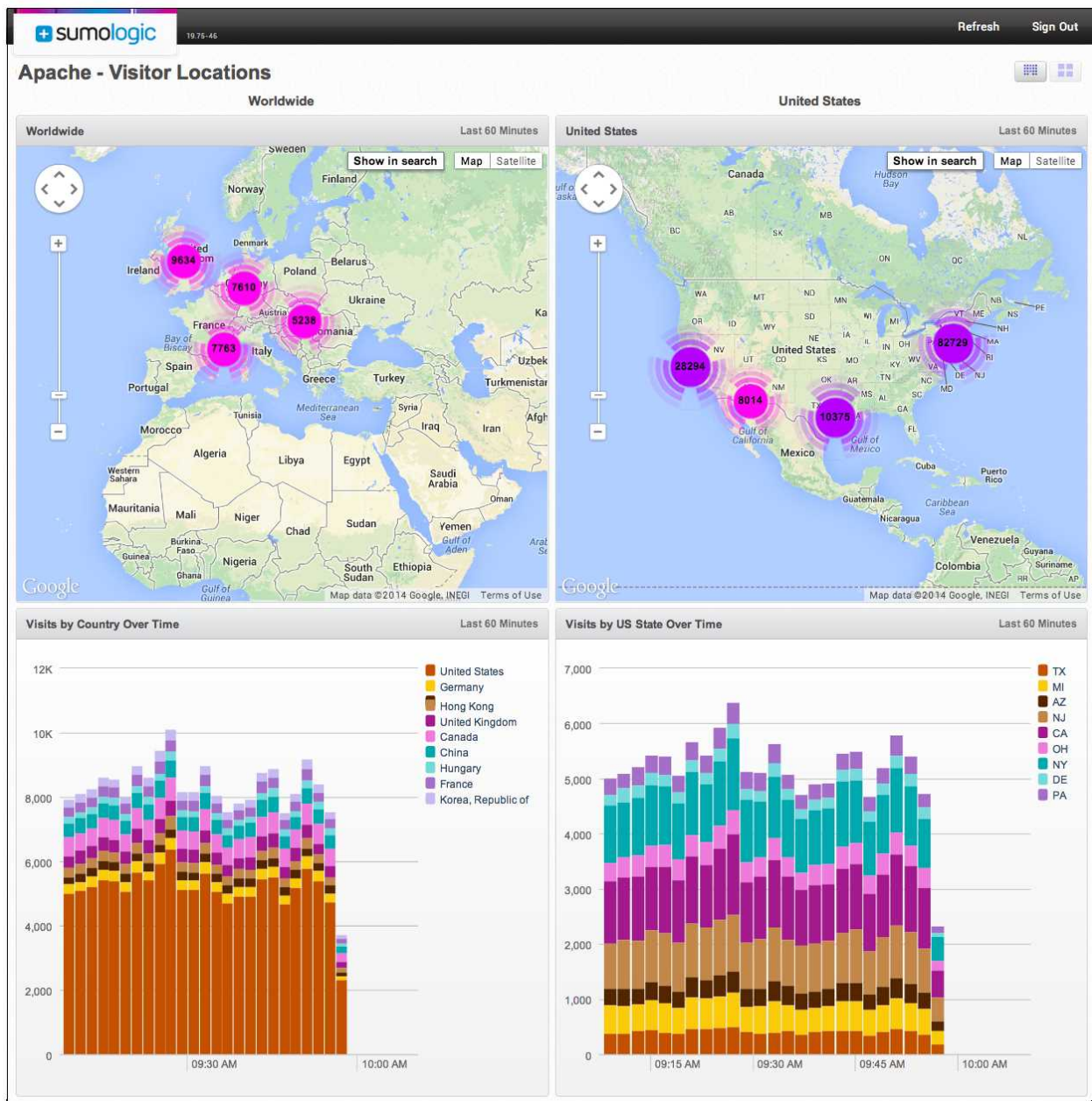
Client Locations. A geolocation query displays where IPs visiting your site originate.

Top Clients. Displays a list of the top IPs visiting your site.

Browsers and Operating Systems. Keep an eye on the browser and OS configurations used to access your site.

Slowest URLs. Displays a list of the five slowest URLs from your site.

Visitor Locations Dashboard



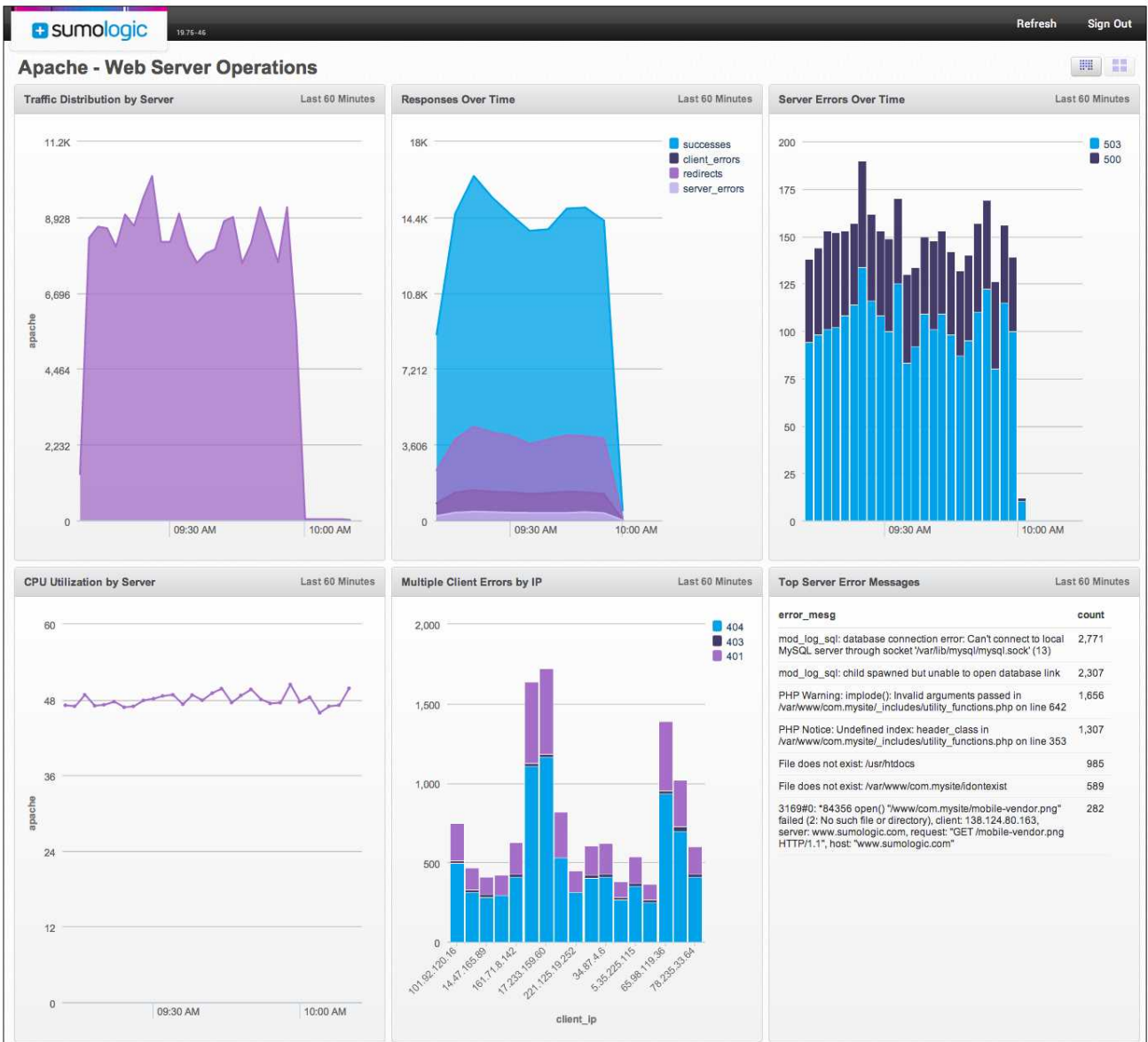
Worldwide. View a map to see where IPs originate.

Visits by Country Over Time. See the worldwide origination of your visitors' IP addresses.

United States. View a map of the US to see where IPs originate.

Visits by US State Over Time. See the US state your visitors' IP addresses originate.

Web Server Operations Dashboard



Traffic Distribution by Server. See which source hosts are handling the load.

CPU Utilization by Server. Keep an eye on the health of your Apache servers.

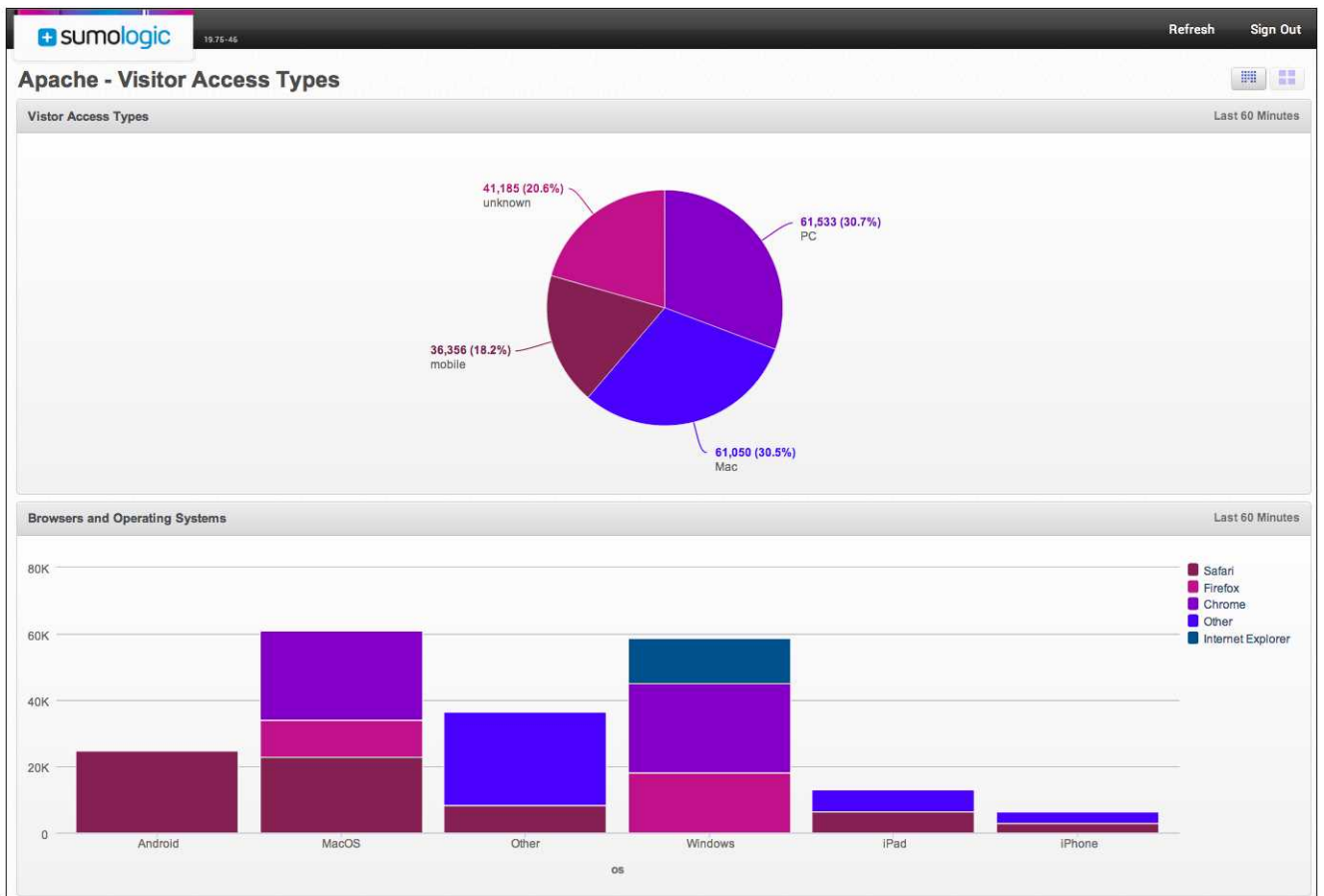
Responses Over Time. Shows the successes, client errors, redirects, and server errors that occurred over the past hour.

Multiple Client Errors by IP. Shows which IP addresses are generating the most 4xx errors.

Server Errors Over Time. See the distribution on 500 and 503 errors over the past hour.

Top Server Error Messages. Displays the counts of the most common errors generated in the past hour.

Visitor Access Types Dashboard



Visitor Access Types. Displays a breakdown of the type of devices visitors are using.

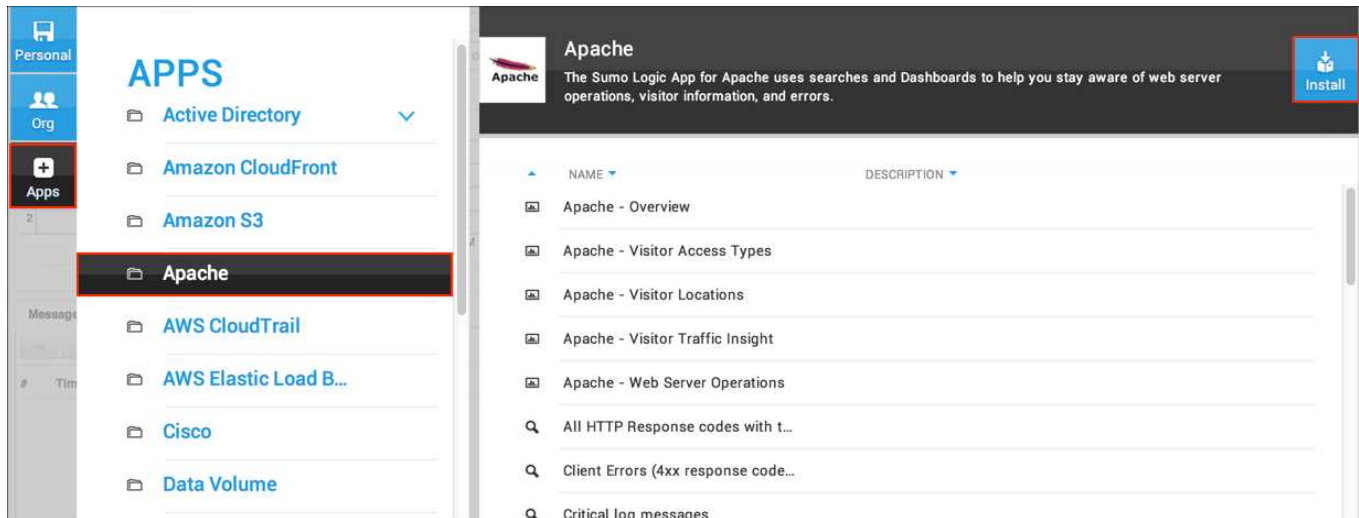
Browsers and Operating Systems. Breaks down the number of users by their browser and OS.

Installing the Sumo Logic App for Apache

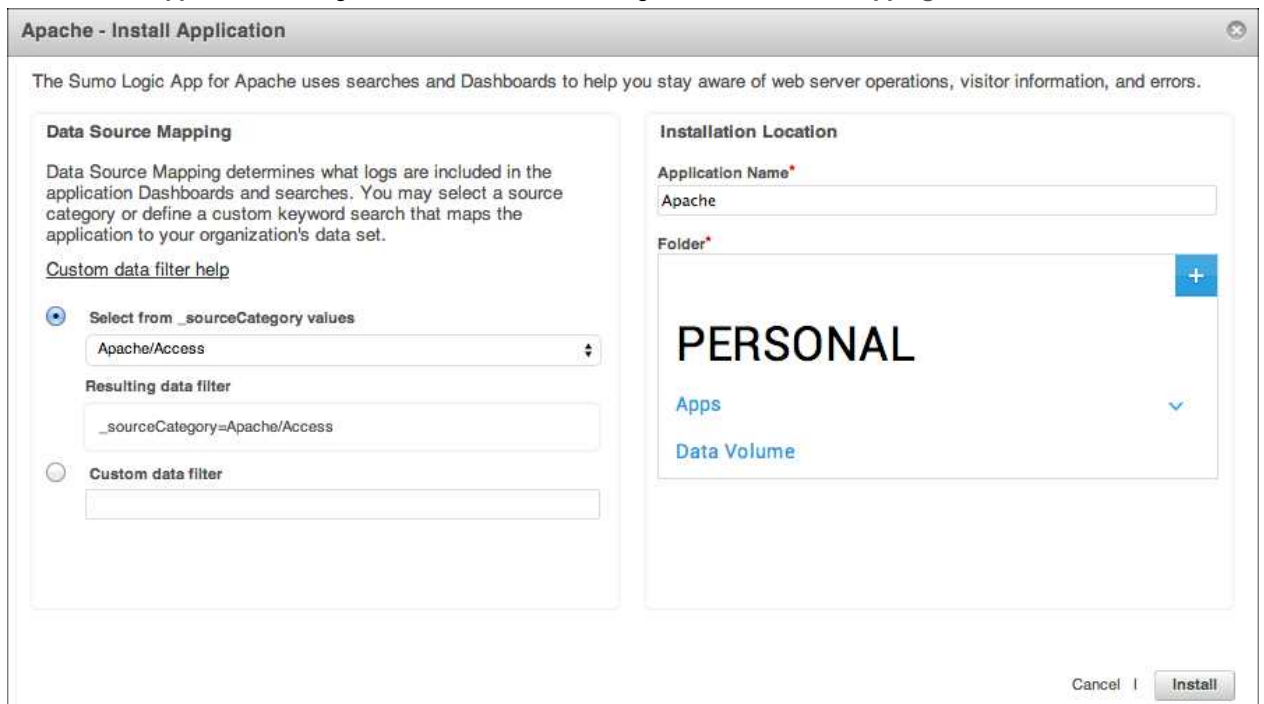
The **Library** feature of the Sumo Logic Web Application allows an Admin to install the Sumo Logic app for Apache. Your organization will be up and running with the app in just a few minutes.

To install the app:

1. In the Library, click the **Apps** tab.
2. Click **Apache**.
3. Click **Install**.



4. In the Install Application dialog box, do one of the following for **Data Source Mapping**:



- **Select from _sourceCategory values.** Choose an existing _sourceCategory present in your account used for the Apache Source, such as **Apache/Access**.
 - To analyze only Apache Access logs, choose a source category that matches the Apache Access logs. A majority of searches and Dashboards in this application are written for Apache Access logs,

so some Dashboard monitors and searches that are based on error logs will not work.

- To monitor only Apache error logs, choose a source category that matches the Apache error logs. The majority of searches and Dashboards in this application are written for Apache Access logs, so most Dashboard monitors and searches that are based on Access logs will not work.
 - **Important:** If you do not select the correct `_sourceCategory`, data will not be loaded into the app. If you don't know which `_sourceCategory` to select, ask the administrator who configured the Source.
 - **Custom data filter.** To set up a specific data filter, type the keyword(s) you'd like to use to filter the data. If you want to analyze both Apache Access logs and error logs, please use a custom data filter that selects both Access and error log data.
5. For **Folder**, choose either the **Personal** folder or a subfolder in the **Personal** folder. (Click the blue + to create a new subfolder).
 6. Click **Install**.

Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

Sumo Logic App for AWS CloudTrail

Amazon Web Services (AWS) CloudTrail records API calls made to AWS. The Sumo Logic App for CloudTrail ingests these logs, providing greater visibility into events that, in turn, allows for security and operations forensics. For example, you can use the Sumo Logic App for CloudTrail to analyze raw CloudTrail data to investigate user behavior patterns. Or, by correlating CloudTrail data with other data sets, you can get a broader understanding of events from operating systems, intrusion detection systems, or even application logs.



Dashboards in the CloudTrail App use a brand new layout that makes certain Monitors larger than others. If you create a new Dashboard or Monitor, it will use the standard layout, where all Monitors in a Dashboard are a uniform size.

Before you begin

Before you can begin to use the Sumo Logic App for CloudTrail, you'll need to make sure that you've configured CloudTrail in your AWS account. Additionally, confirm that logs are being delivered to the S3 Bucket you'll use to send the logs to Sumo Logic. For more information, and instructions, see [Collecting logs for the Sumo Logic for CloudTrail App](#).

Using the App for CloudTrail in multiple environments

If you have more than one environment that generates CloudTrail data (such as ops, dev, and so on) you'll need to configure a separate S3 Source for each environment. You can learn more [here](#).

Sumo Logic App for AWS CloudTrail Dashboards

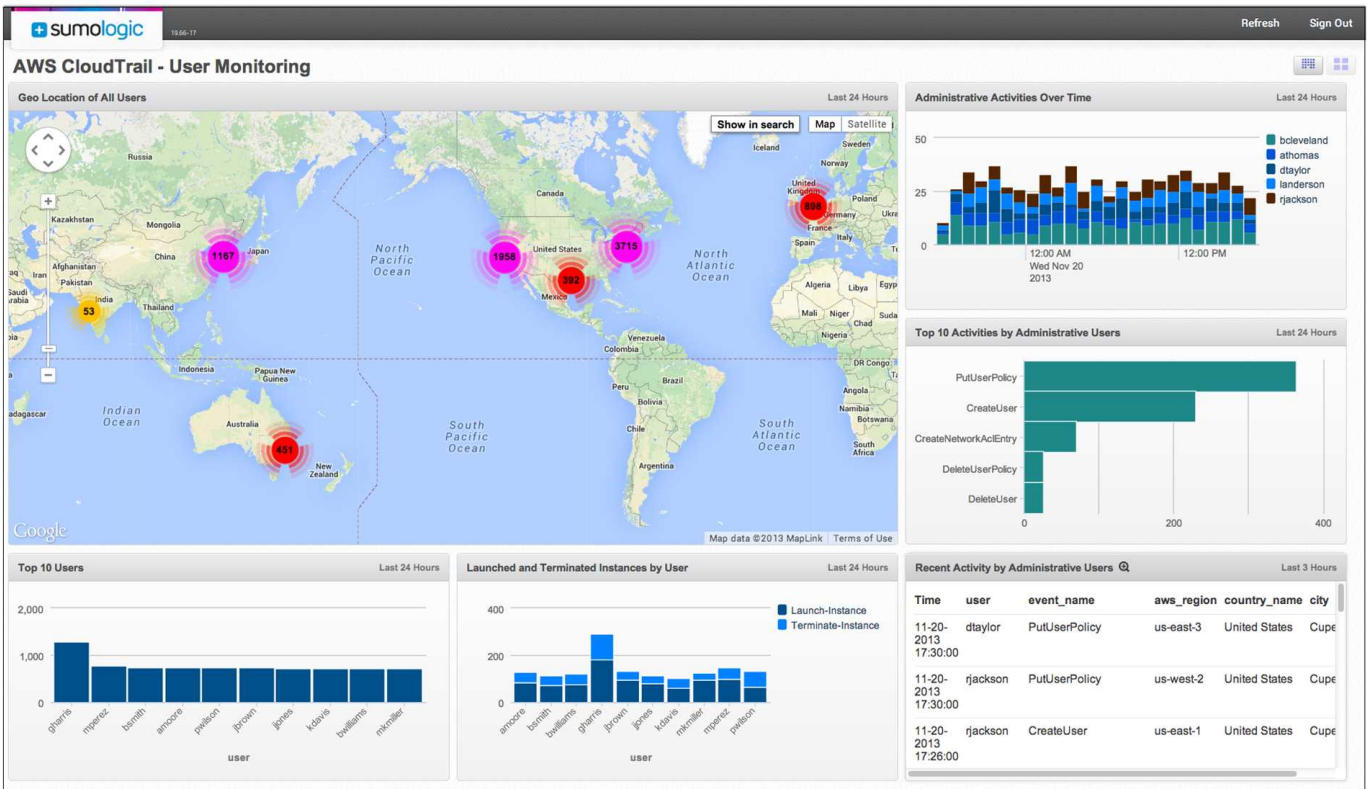
The Sumo Logic App for AWS CloudTrail has three Dashboards that keep an eye on user activity, operations, and network and security events.

What if data isn't displaying in all Monitors?

Amazon S3 buckets are scanned for new files according to the Scan Interval you set when configuring the S3 Source used for AWS CloudTrail logs. Even if you set a shorter Scan Interval, say five minutes, if no new files are found, the Scan Interval is automatically doubled, up to 24 hours (you can read more in [Setting the S3 Source Scan Interval](#)). If the Scan Interval increases, it means that a Monitor set to a 60 minute time range may not find any data to display, because no files have uploaded to Sumo Logic. This isn't to say that no data is being collected from your S3 bucket; you can confirm that data is being collected on the Status page.

Additionally, you can change the time range of a Monitor. Even though these Monitors have been preconfigured, they can be edited just like any other Monitor. You'll find instructions in [Changing the time range of a Monitor](#).

User Monitoring Dashboard



Geo Location of All Users. Using a geo location search, this Monitor shows the locations of the IPs used by visitors.

Administrative Activities Over Time. Shows which administrative users have been active every hour over the past 24 hours.

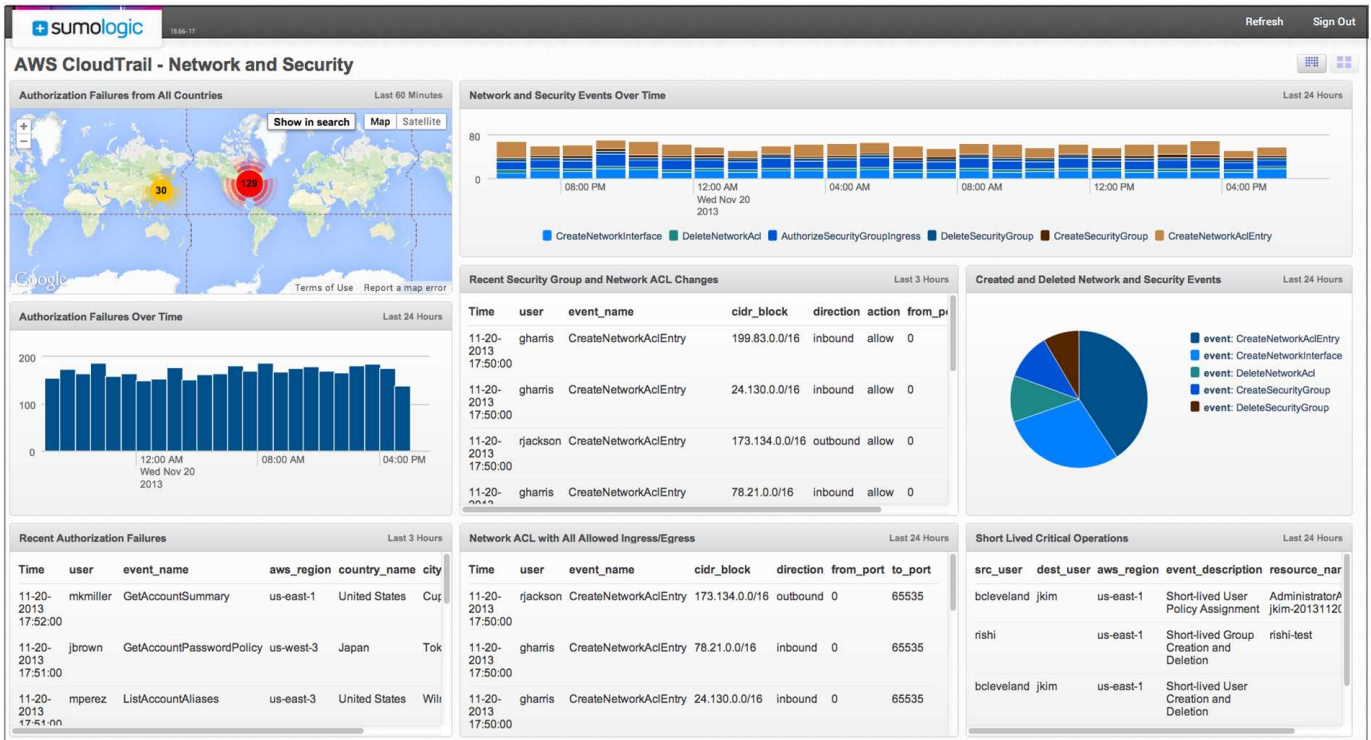
Top 10 Activities by Administrative Users. See which activities have been performed the most by administrative users.

Recent Activity by Administrative Users. Activity over the last three hours are displayed by the name of the event (CreateUser, PutUserPolicy, and so on) and by the user's name and location.

Launched and Terminated Instances by User. Shows the number of instances that have either been launched or terminated every hour over the past 24 hours.

Top 10 Users. This Monitor displays the top 10 most active AWS users.

Network and Security Dashboard



Authorization Failures from All Countries. Uses a geolocation search to display a map of where failures occur world-wide.

Authorization Failures Over Time. View the number of “Access Denied” errors generated every hour over the past 24 hours.

Recent Authorization Failures. Shows the most recent authorization failures.

Network and Security Events Over Time. Displays the number of specific events every hour over the past 24 hours.

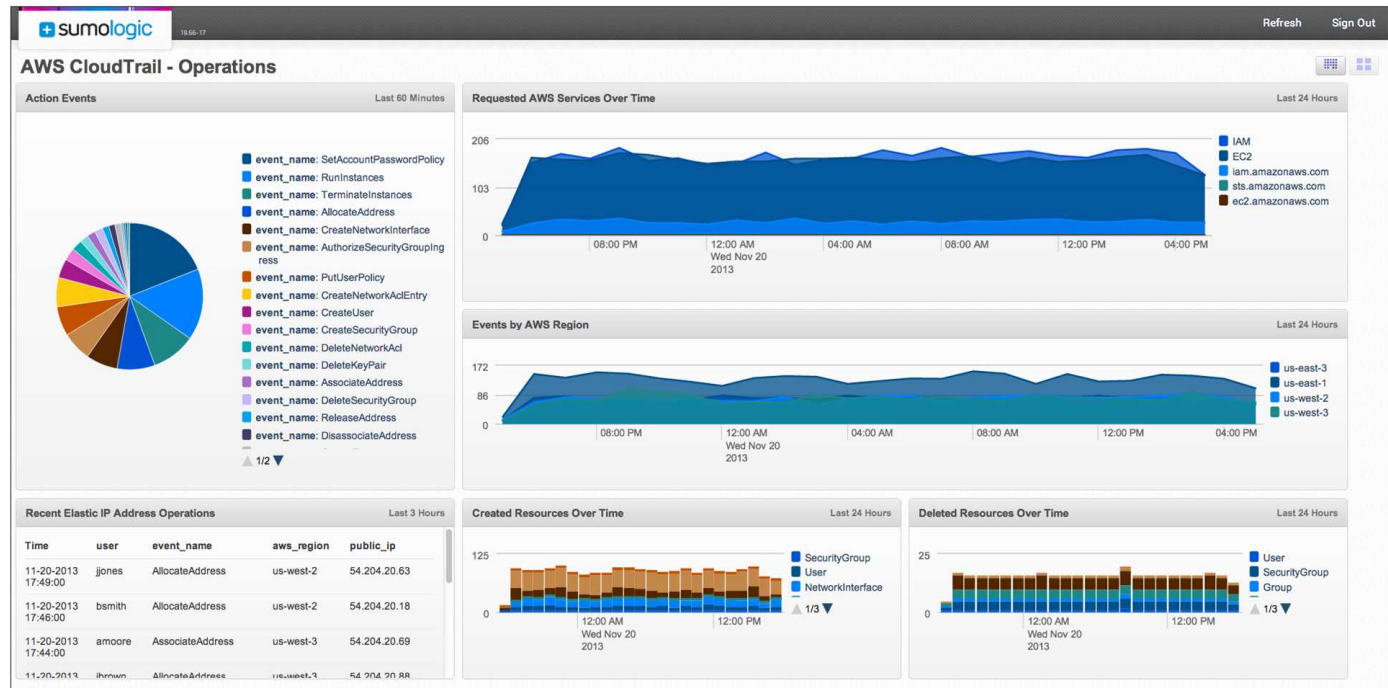
Recent Security Group and Network ACL Changes. Shows the most recent changes that were made to security groups in the form of authorizing ingress to a security group or the creation of a network access control list over the past three hours.

Network ACL with All Allowed Ingress/Egress. Displays a list all of inbound or outbound events where ingress or egress for a particular subnet was allowed for all possible ports.

Created and Deleted Network and Security Events. Displays a chart of created or deleted events.

Short Lived Critical Operations. The search behind this Monitor watches for users, groups, or policies that are created and then deleted within a span of 10 minutes

Operations Dashboard



Action Events. Displays a list of events that correspond to a user performing a certain AWS action over the past hour.

Requested AWS Services Over Time. Shows the number of requests every hour over the past 24 hours for AWS services, like EC2 and IAM.

Events by AWS Region. Makes it easy to watch the number of events in each AWS region every hour over the past 24 hours.

Recent Elastic IP Address Operations. View the most recent operations (from the past three hours), displayed by IP address, user, and AWS region.

Created Resources Over Time. Displays a day's worth of created resources every hour across your deployment.

Deleted Resources Over Time. Displays the resources deleted every hour over the past 24 hours.

Collecting logs for the Sumo Logic App for AWS CloudTrail

Before you can begin to use the Sumo Logic App for CloudTrail, you'll need to make sure that you've configured CloudTrail in your AWS account. Additionally, confirm that logs are being delivered to the S3 Bucket you'll use to send the logs to Sumo Logic.



Once you begin uploading CloudTrail data your daily data usage will increase. It's a good idea to check the Account page in the Sumo Logic Web Application to make sure that you have enough quota to accommodate additional data in your account. If you need additional quota you can upgrade at any time.

Configuring a Hosted Collector and an S3 Source

You'll create a new Hosted Collector and an S3 Source that will be used to upload the CloudTrail data to Sumo Logic.



Although logs from multiple AWS accounts may be associated with a single S3 bucket within your organization, it's recommended that you configure one S3 Source for each account. This allows for easier searching and analysis of your data.

To collect CloudTrail logs:

1. In the Sumo Logic Web Application, create a new [Hosted Collector](#).
2. Next, [configure an S3 Source](#) for the S3 bucket using the following settings:
 - Type the **Name** for the Source. It's a good idea to use a Source Name that reflects the purpose of the AWS account; for example, BillingApplication.
 - For **Bucket Name**, type the *exact* name of the S3 bucket.
 - For **Path Expression**, type the path where your CloudTrail logs reside, starting with the prefix. Do not enter a leading forward slash. For example, `aws/AWSLogs/<account ID>/CloudTrail/*`, making sure to replace `<account ID>` with your 12-digit account number. Note that there is no leading forward slash.
 - Type **AWS_EAGLE** in the Source Category text box. **Important: Do not type any other text for Source Category in order to use the CloudTrail app.**
 - For **Key ID** and **Secret Key**, type the values generated in AWS. (You'll find more information in [Granting Access to an S3 bucket](#).)
 - For **Scan Interval**, choose an option to set the frequency of when the S3 bucket will be scanned for new logs. No matter the interval you choose, if no new logs are found during a scan, the interval will double, up to 24 hours. This means that some Monitors in the AWS CloudTrail app may not display data as you'd expect, depending on the time range of the Monitor. You can learn more in [Setting the S3 Scan Interval](#).

Amazon S3
Collects logs from an Amazon S3 bucket.

HTTP
HTTP receiver that collects logs sent to a specific address.

Name* BillingApplication

Description

Bucket Name* OurBucketName

Path Expression* aws/AWSLogs/123456789012/*
Path expression to match one or more S3 objects.
For example: ABC*.log or ABC.log

Collection should begin 24 hours ago
(starts approx. at 11/20/2013 10AM)

Source Category AWS_EAGLE
Log category metadata to use later for querying, e.g. OS_Security

Key ID*
Your AWS Access Key ID

Secret Key*
Your AWS Secret Access Key

Scan Interval* 5 Minute(s)
The frequency in which the source scans the S3 Bucket. Setting this value too low may incur additional charges by Amazon.

► Advanced

► Filters

Save | **Cancel**

3. For the rest of the settings, the default options can be used.
4. Click **Save**.

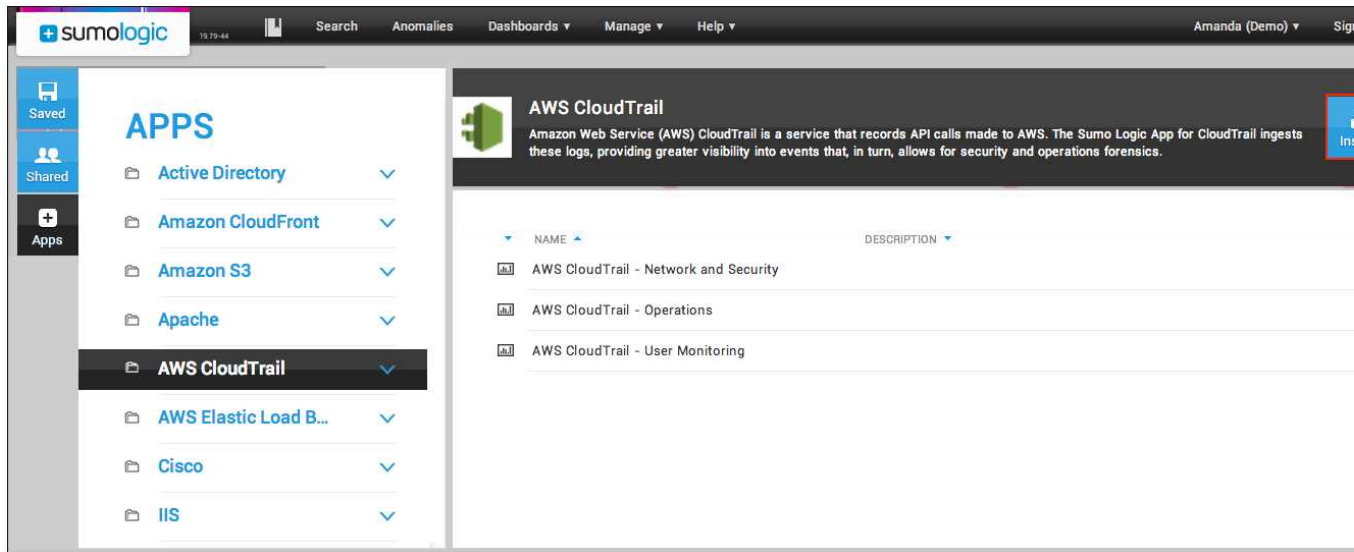
Installing and starting the AWS CloudTrail App

The Library feature of the Sumo LogicWeb Application allows an Admin to install the QuickStart App. Your organization will be up and running with the app in just a few minutes.

To install the app:

1. In the **Library**, click the **Apps** tab.
2. Click **AWS CloudTrail**.

3. Click Install.



4. In the Install Application dialog box, enter a Custom data filter if desired.
5. Choose a location in the Personal folder.
6. Click Install.

AWS CloudTrail - Install Application

Amazon Web Service (AWS) CloudTrail is a service that records API calls made to AWS. The Sumo Logic App for CloudTrail ingests these logs, providing greater visibility into events that, in turn, allows for security and operations forensics.

Data Source Mapping

Data Source Mapping determines what logs are included in the application Dashboards and searches. You may select a source category or define a custom keyword search that maps the application to your organization's data set.

[Custom data filter help](#)

For the AWS CloudTrail application, the sourceCategory is pre-set to AWS_EAGLE and may not be changed. You may optionally specify an additional custom source.

Custom data filter

Installation Location

Application Name*

Folder*

PERSONAL

Cancel | **Install**

Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

Configuring the App for AWS CloudTrail in multiple environments

If you have more than one environment that generates CloudTrail data (such as ops, dev, and so on) you'll need to configure a separate S3 Source for each environment, meaning that you'll have the three App Dashboards for each environment.

To avoid confusion, and in order to identify which environment is generating data, you should name each S3 Source with the environment's name. For example, you might name Sources as:

- CloudTrail-prod
- CloudTrail-dev
- CloudTrail-test
- and so on

Finally, make copies of each Monitor in the CloudTrail Dashboards, and modify the search logic in each Monitor so that you select the appropriate source for each environment.

For example, for a production environment, you will add the string: **`_source=CloudTrail-production`** to the beginning of each search. Edit the names of the Monitors as needed. This means if you have three environments then you will have three copies of the application for each of them (nine dashboards in total).

Sumo Logic App for AWS CloudFront

Amazon Web Services (AWS) CloudFront is a content delivery network (CDN) that allows an easy way for companies to distribute content to end-users with low latency and high data transfer speeds. The Sumo Logic Application for CloudFront provides analytics on visitor information, rates and statistics, content being served, and other metrics. The app uses predefined searches and Dashboards that provide visibility into your environment for real time analysis of overall usage.

Visitor Statistics Dashboard



Client Geo Distribution. Performs a geo lookup search and displays visitor's client distribution for the last 24 hours on a map of the world.

Requests Served by Edge Location. Displays visitor requests served by edge location sorted by count for the last three hours in a pie chart.

Visitor Access Platforms. Provides information on the platforms that visitors use to access the site for the last three hours in a pie chart.

Visitor Session Duration Distribution Histogram. Displays the duration of visitor sessions distributed by count and bucket size in a histogram.

Unique Visitors Over Time. Shows unique visitors to the site by based on IP address in timeslices of five minutes over the last three hours in a column chart.

Visitor Browsers and Devices. Displays the devices and browsers, counted by platform, used by visitors to access the site over the last three hours in a stacked column chart.

Web Operations Dashboard



Edge Result. Displays edge results by count and sorted by type for the last three hours in a pie chart.

Client and Server Errors Over Time. Shows client and server errors over time in timeslices of five minutes for the last three hours in a column chart.

HTTP Response Classes. Provides HTTP response classes by count in timeslices of five minutes for the last three hours in a timeline.

Cache Hit and Miss Over Time. Displays the cache's hits and misses over time in timeslices of five minutes for the last three hours in a stacked column chart.

HTTP Status Codes Over Time. Shows HTTP status codes over time in timeslices of five minutes for the last three hours in a timeline.

Traffic and Megabytes Served. Provides information on site traffic hits and Megabytes served in timeslices of one hour over the last 24 hours in a combination column and line chart.

Sumo Logic App for AWS Elastic Load Balancing

Amazon Web Services' (AWS) Elastic Load Balancing distributes incoming application traffic across multiple Amazon EC2 instances in the AWS Cloud. The Sumo Logic App for Elastic Load Balancing ingests logs generated by this activity, providing greater visibility into events that, in turn, help you understand the overall health of your EC2 deployment. For example, you can use the Sumo Logic App to analyze raw Elastic Load Balancing data to investigate the availability of applications running behind Elastic Load Balancers. Or, by correlating Elastic Load Balancing data with other data sets, you can get a broader understanding of the fault tolerance of your applications across multiple AWS Availability Zones.

Collecting logs for the AWS Elastic Load Balancing App

Before you can begin to use the Sumo Logic App for Elastic Load Balancing, you'll need to enable logging in AWS, and configure an S3 bucket to collect Elastic Load Balancing logs in your AWS account. Then, you must confirm that logs are being delivered to the S3 bucket that sends the logs to Sumo Logic.

Once you begin uploading Elastic Load Balancing data, your daily data usage will increase. It's a good idea to check the **Account** page in the Sumo Logic Web Application to make sure that you have enough quota to accommodate additional data in your account. If you need additional quota you can upgrade at any time.

Enable logging in AWS

By default logging is not enabled for Elastic Load Balancers. You can find additional assistance for enabling logging in AWS Documentation.

To enable logging in AWS:

1. In the AWS Management Console, choose **EC2 > Load Balancers**.
2. Under Access Logs click **Edit**.
3. In the Configure Access Logs dialog box click **Enable Access Logs**, then choose an Interval and S3 bucket. This is the S3 bucket that will upload logs to Sumo Logic.
4. Click **Save**.

After setting up the S3 bucket check to make sure logs are being delivered.

Configuring a Hosted Collector and S3 Source

After confirming that logs are being delivered to the S3 bucket, it's time to create a new Hosted Collector and an S3 Source that will be used to upload the Elastic Load Balancing data to Sumo Logic.



Although logs from multiple AWS accounts may be associated with a single S3 bucket within your organization, it's recommended that you configure one S3 Source for each account. This allows for easier searching and analysis of your data.

To collect Elastic Load Balancing logs:

1. In the Sumo Logic Web Application, create a new [Hosted Collector](#).
2. Next, configure an S3 Source for the S3 bucket using the following settings:
 - Type the Name for the Source. It's a good idea to use a Source Name that reflects the purpose of the AWS account; for example, BillingApplication.
 - For **Bucket Name**, type the exact name of the S3 bucket.
 - For **Path Expression**, type the path where your Elastic Load Balancing logs reside, starting with the prefix. Do not enter a leading forward slash. For example, aws/AWSLogs/<account ID>/ElasticLoadBalancing/*, making sure to replace <account ID> with your 12-digit account number. Note that there is no leading forward slash.
 - Type **AWS_ELB** in the Source Category text box.
Important: Do not type any other text for Source Category in order to use the Elastic Load Balancing app.
 - For **Key ID** and **Secret Key**, type the values generated in AWS.
 - For **Scan Interval**, choose an option to set the frequency of when the S3 bucket will be scanned for new logs. No matter the interval you choose, if no new logs are found during a scan, the interval will double, up to 24 hours. This means that some Monitors in the AWS CloudTrail app may not display data as you'd expect, depending on the time range of the Monitor. You can learn more in [Setting the S3 Scan Interval](#).
 - For the rest of the settings, the default options can be used.
3. Click **Save**.

(Optional) Configuring the Sumo Logic App for Elastic Load Balancing in Multiple Environments

If you have more than one environment that generates Elastic Load Balancing data (such as ops, dev, and so on) you'll need to configure a separate S3 Source for each environment. This means that you'll have the three App Dashboards for each environment. To avoid confusion, and in order to identify which environment is generating data, you should name each S3 Source with the environment's name. For example, you might name Sources as ELB-prod, ELB-dev, ELB-test, etc.

Finally, make copies of each Monitor in the Elastic Load Balancing Dashboards, and modify the search logic in each Monitor so that you select the appropriate source for each environment. For example, for a production environment, you will add the string: `_source=ELB-production` to the beginning of each search. Instructions for editing the name of a Monitor can be found [here](#). This means if you have three environments then you will have three copies of the application for each of them (nine dashboards in total).

AWS Elastic Load Balancing Dashboards

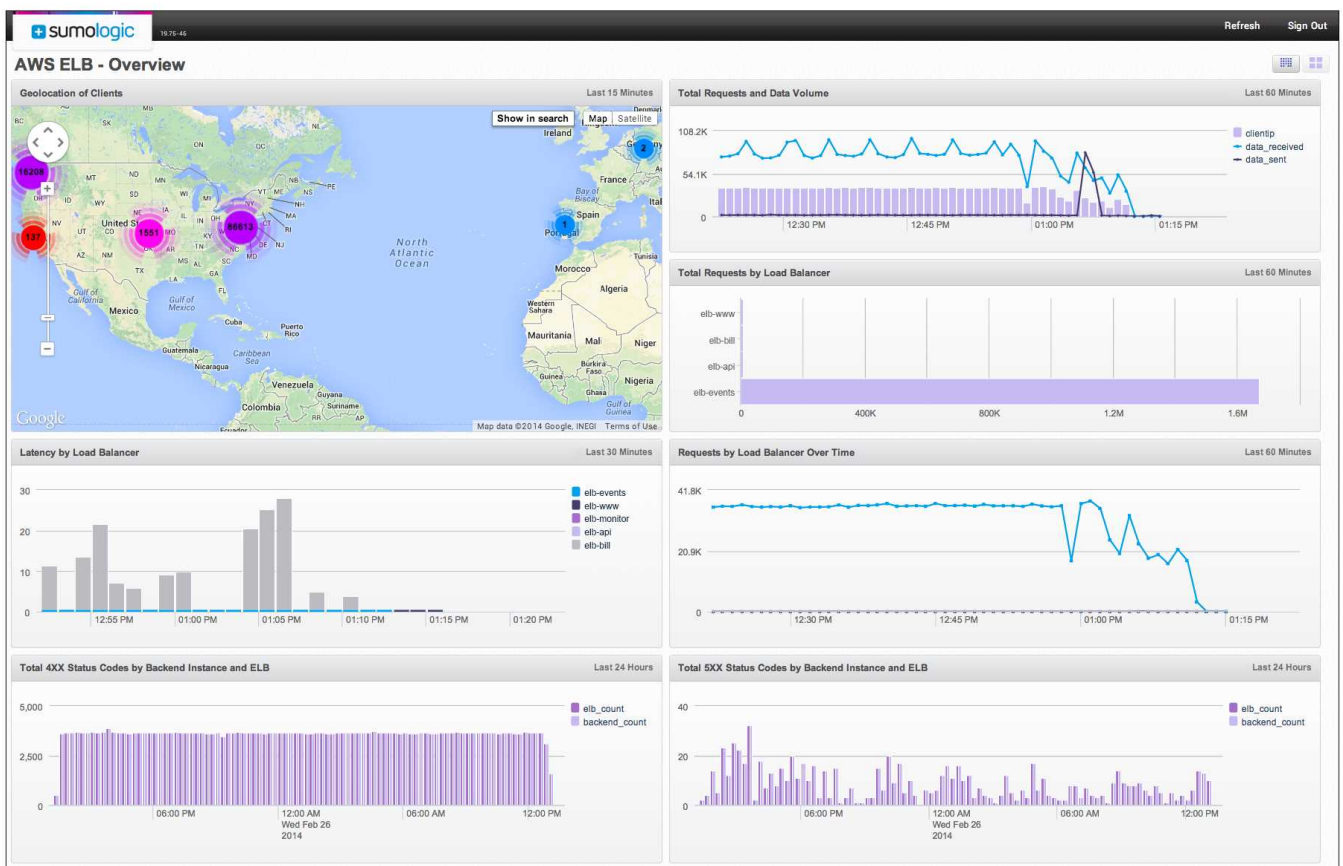
The Sumo Logic App for AWS Elastic Load Balancing has three Dashboards. In addition to helping you monitor the overall health of your ELB deployment, Dashboards keep an eye on errors being generated by back-end applications as well as errors generated from ELB instances.

What if data isn't displaying in all Monitors?

Amazon S3 buckets are scanned for new files according to the Scan Interval you set when configuring the S3 Source used for AWS Elastic Load Balancing logs. Even if you set a shorter Scan Interval, say five minutes, if no new files are found, the Scan Interval is automatically doubled, up to 24 hours (you can read more in [Setting the S3 Source Scan Interval](#)). If the Scan Interval increases, it means that a Monitor set to a 60 minute time range may not find any data to display, because no files have uploaded to Sumo Logic. This isn't to say that no data is being collected from your S3 bucket; you can confirm that data is being collected on the Status page.

Additionally, you can change the time range of a Monitor. Even though these Monitors have been preconfigured, they can be edited just like any other Monitor. You'll find instructions in [Changing the time range of a Monitor](#).

Overview Dashboard



Geolocation of Clients. Uses a geolocation query to display a map of the IP addresses used by clients accessing your apps.

Latency by Load Balance. Displays the latency of each load balancer in AWS, over time.

Total 4XX Status Codes by Backend Instance and ELB. Charts the number of 4XX status codes for each backend instance and ELB.

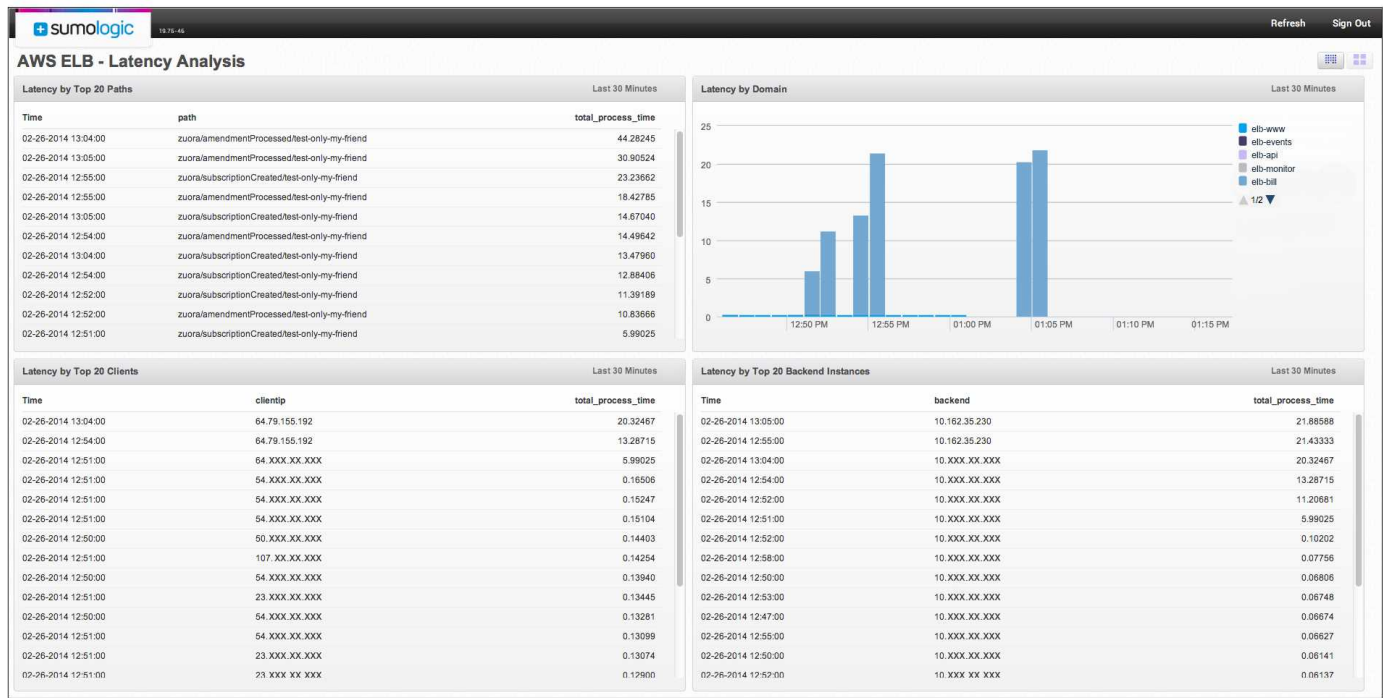
Total Requests and Data Volume. Displays the data being sent and received by client IP.

Total Requests by Load Balancer. Show the requests per load balancer over time.

Requests by Load Balancer Over Time. Displays a line chart of how many requests are hitting a load balancer over time.

Total 5X Status Codes by Backend Instance and ELB. Displays a breakdown of 5XX errors occuring in your ELB instances and back end servers.

Latency Analysis Dashboard



Latency by Top 20 Paths. Displays the process time of the busiest 20 paths in your deployment.

Latency by Top 20 Clients. Displays the process time of the top 20 IP addresses in your deployment.

Latency by Domain. Displays the latency of each domain in your EC2 deployment.

Latency by Top 20 Backend Instances. This Monitor focuses solely on the back end of your AWS EC2 deployment, keeping watch over the processing time of the 20 busiest instances.

Status Codes Analysis Dashboard

sumologic

19.75-46

RefreshSign Out

AWS ELB - Status Codes Analysis

Backend Instance and Load Balancer 4XX Status Codes by PathLast 30 Minutes

Time	path	load_balancer	backend_instance
02-26-2014 12:50:00	receiver/v1/http/ZaVn/C4dhaV15U0XJCRFAad6ZWZ0UdoB1GG3UmgZaVLZRmOD04FQLb6FNjXmL4PipISvAYh2RZAgCaV9ITEp-ZahZFrEXSJaabjE4BGa4tDOAILU7A==	236	
02-26-2014 12:51:00	receiver/v1/http/ZaVn/C4dhaV15U0XJCRFAad6ZWZ0UdoB1GG3UmgZaVLZRmOD04FQLb6FNjXmL4PipISvAYh2RZAgCaV9ITEp-ZahZFrEXSJaabjE4BGa4tDOAILU7A==	190	
02-26-2014 12:54:00	zuora/subscriptionCreated/test-only-my-friend	9	
02-26-2014 12:54:00	zuora/subscriptionCreated/test-only-my-friend	7	

Backend Instance and Load Balancer 4XX Status Codes by DomainLast 30 Minutes

Time	domain	load_balancer	backend_instance	total_4xx
02-26-2014 12:50:00	etb-events.net	236	236	472
02-26-2014 12:51:00	etb-events.net	167	167	334
02-26-2014 12:54:00	etb-bill.net	9	9	18
02-26-2014 13:04:00	etb-bill.net	7	7	14
02-26-2014 12:55:00	etb-bill.net	5	5	10
02-26-2014 13:05:00	etb-bill.net	5	5	10
02-26-2014 12:52:00	etb-bill.net	4	4	8
02-26-2014 12:52:00	etb-events.net	1	1	2

Backend Instance and Load Balancer 4XX Status Codes by ClientLast 30 Minutes

Time	clientip	load_balancer	backend_instance	total_4xx
02-26-2014 12:51:00	12.177.21.34	240	240	480
02-26-2014 12:50:00	12.177.21.34	236	236	472
02-26-2014 12:52:00	12.XXX.XX.XX	49	49	98
02-26-2014 12:54:00	64.XXX.XX.XX	9	9	18
02-26-2014 13:04:00	64.XXX.XX.XX	7	7	14
02-26-2014 13:05:00	64.XXX.XX.XX	5	5	10
02-26-2014 12:55:00	64.XXX.XX.XX	5	5	10
02-26-2014 12:52:00	64.XXX.XX.XX	4	4	8

Backend Instance and Load Balancer 5XX Status Codes by PathLast 30 Minutes

Time	path	load_balancer	backend_instance	total_5xx
02-26-2014 13:05:00	zuora/amendmentProcessed/test-only-my-friend	4	4	8
02-26-2014 12:55:00	zuora/amendmentProcessed/test-only-my-friend	3	3	6
02-26-2014 12:54:00	zuora/amendmentProcessed/test-only-my-friend	3	3	6
02-26-2014 13:04:00	zuora/amendmentProcessed/test-only-my-friend	2	2	4
02-26-2014 12:52:00	zuora/amendmentProcessed/test-only-my-friend	2	2	4

Backend Instance and Load Balancer 5XX Status Codes by DomainLast 30 Minutes

Time	domain	load_balancer	backend_instance	total_5xx
02-26-2014 13:05:00	etb-bill.net	4	4	8
02-26-2014 12:55:00	etb-bill.net	3	3	6
02-26-2014 12:54:00	etb-bill.net	3	3	6
02-26-2014 13:04:00	etb-bill.net	2	2	4
02-26-2014 12:52:00	etb-bill.net	2	2	4

Backend Instance and Load Balancer 5XX Status Codes by ClientLast 30 Minutes

Time	clientip	load_balancer	backend_instance	total_5xx
02-26-2014 13:05:00	64.79.155.192	4	4	8
02-26-2014 12:54:00	64.79.155.192	3	3	6
02-26-2014 12:55:00	64.XX.XXX.XXX	3	3	6
02-26-2014 13:04:00	64.XX.XXX.XXX	2	2	4
02-26-2014 12:52:00	64.XX.XXX.XXX	2	2	4

Backend Instance and Load Balancer 4XX Status Codes by Path. Displays the time an error occurred, along with the associated path.

Backend Instance and Load Balancer 5XX Status Codes by Path. Shows the time error(s) occurred on a specific path, along with the load balancer and backed instance associated with the path.

Backend Instance and Load Balancer 4XX Status Codes by Domain. Displays the time of an error, along with the domain, the load balancer associated with the domain, and the error code.

Backend Instance and Load Balancer 5XX Status Codes by Domain. Shows the time error(s) occurred in a domain, along with the load balancer and backed instance associated with domain.

Backend Instance and Load Balancer 4XX Status Codes by Client. Shows the time an error occurred, the IP that generated the error, the load balancer associated with the UP, and the number of errors that have occurred.

Backend Instance and Load Balancer 5XX Status Codes by Client. Shows the time error(s) occurred at an IP, along with the load balancer and backed instance associated with the client IP.

Sumo Logic App for Cisco

The Sumo Logic Application for Cisco gives you insight into website visitor patterns, monitors infrastructure operations, and provides easy access to threat monitoring. The app uses a predefined parser, searches, and Dashboard, which provide visibility into your environment for real time analysis of overall usage and threats. The Sumo Logic App for Cisco consists of three main categories: connection statistics, outbound connections, and denied connections.

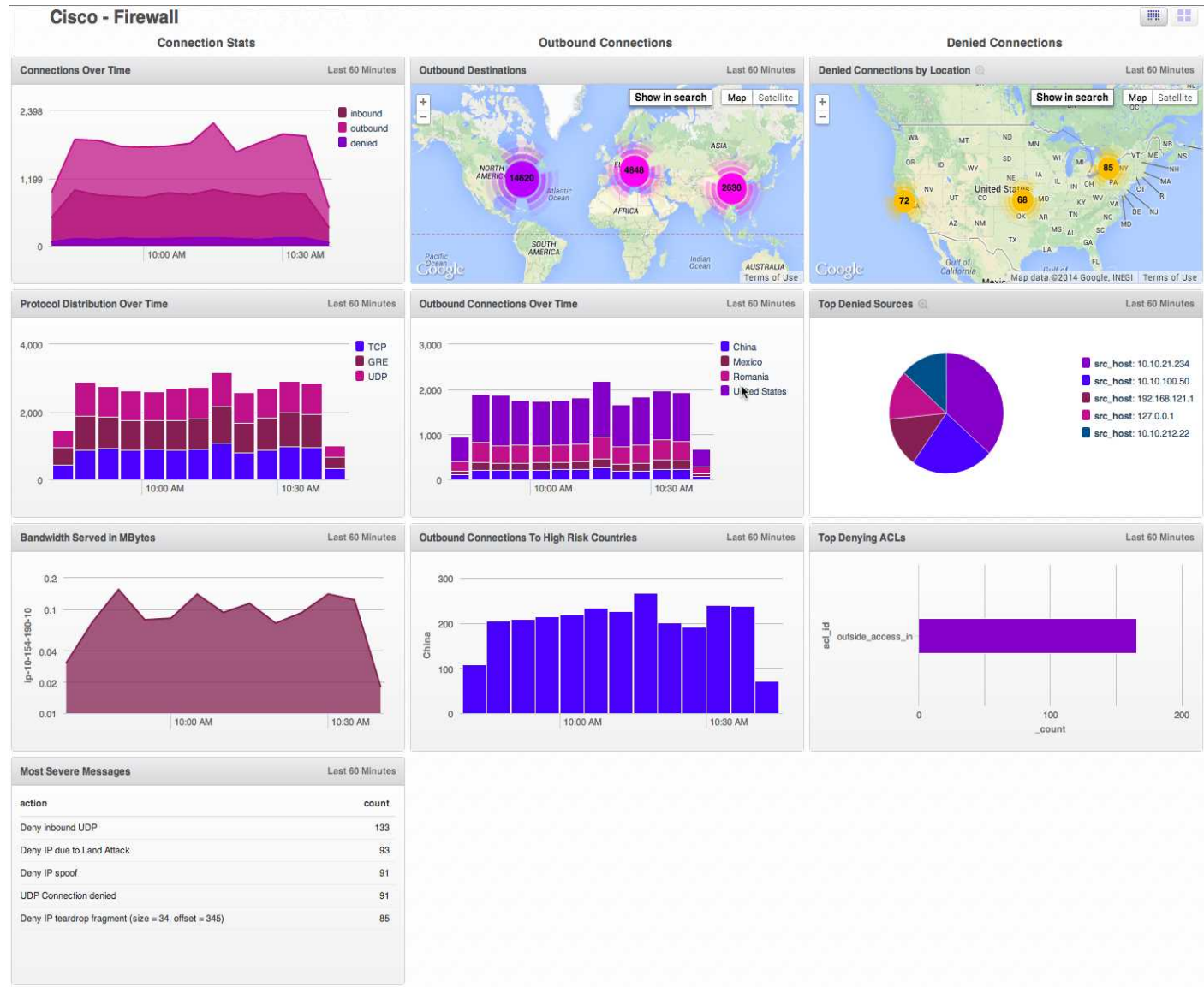
Log Types

The Sumo Logic app for Cisco assumes logs from the Cisco Adaptive Security Appliance (ASA) firewall product.

Sumo Logic App for Cisco Dashboard

The Sumo Logic App for Cisco consists of three main categories: connection statistics, outbound connections, and denied connections based on logs from Cisco ASA.

General Dashboard



Connections Over Time. Provides information on inbound, outbound, and denied connections over the past hour in increments of five minutes in a stacked area chart.

Protocol Distribution Over Time. Displays data on the connection protocols being used, including GRE, UDP, and TCP, over the past hour in increments of five minutes. Displayed in a column chart.

Bandwidth Served in MBytes. Shows the bandwidth served in megabytes over the last hour in five minute increments.

Most Severe Messages. Displays the top five most severe message alerts and their count over the last hour.

Outbound Destinations. A geolocation query tracks the number of outbound connection and displays their destinations on a map of the world. Results are displayed for the last hour. Click **Show in Search** to see more details of the query results.

Outbound Connections Over Time. Provides details on the number of outbound connections by country over the past hour in increments of five minutes.

Outbound Connections to High Risk Countries. Displays the number of outbound connections by country to countries considered high risk over the last hour in five minute increments.

Denied Connections by Location. Uses a geolocation query to track the number of denied connections, and displays their destinations on a map of the world. Results are displayed for the last hour. Click **Show in Search** to see more details of the query results.

Top Denied Sources. Lists the top five denied sources by IP address over the last hour.

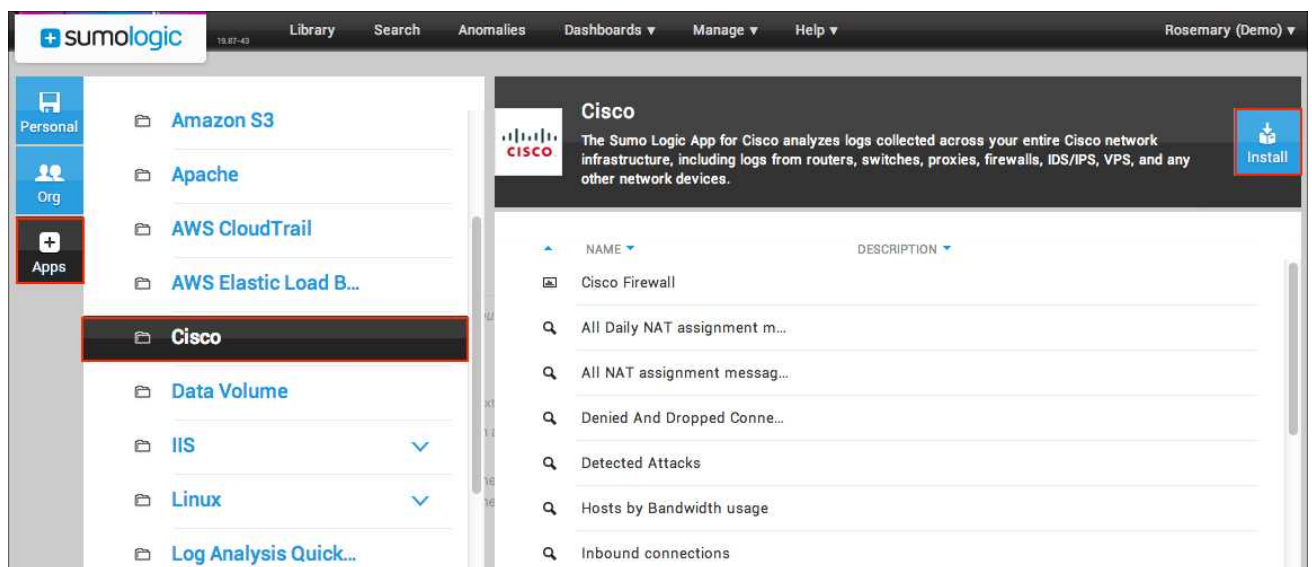
Top Denying ACLs. Displays the top ten denying ACL connections over the last hour in a bar chart.

Installing the Cisco App

The **Library** makes it easy to install the Sumo Logic Applications your organization needs.

To install the App:

1. In the **Library**, click the **Apps** tab.
2. Click **Cisco**.
3. Click **Install**.



4. In the **Install Application** dialog box, do one of the following for **Data Source Mapping**:

Cisco - Install Application

The Sumo Logic App for Cisco analyzes logs collected across your entire Cisco network infrastructure, including logs from routers, switches, proxies, firewalls, IDS/IPS, VPS, and any other network devices.

Data Source Mapping

Data Source Mapping determines what logs are included in the application Dashboards and searches. You may select a source category or define a custom keyword search that maps the application to your organization's data set.

[Custom data filter help](#)

☒ **Select from `_sourceCategory` values**

Cisco/ASA

Resulting data filter

`_sourceCategory=Cisco/ASA`

☐ **Custom data filter**

Installation Location

Application Name*

Cisco

Folder*

PERSONAL

Apache

Cancel | Install

- **Select from `_sourceCategory` values.** Choose an existing `_sourceCategory` present in your account that is associated with logs from Cisco ASA.
Important: If you do not select the correct `_sourceCategory`, data will not be loaded into the app. If you don't know which `_sourceCategory` to select, ask the administrator who configured the Source.
 - **Custom data filter.** To set up a specific data filter, type the keyword(s) you'd like to use to filter the data.
5. For **Folder**, choose either the **Personal** folder or a subfolder in the **Personal** folder. (Click the blue + to create a new subfolder).
6. Click **Install**.

Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

Sumo Logic App for IIS

The Sumo Logic application for IIS allows you to manage your Microsoft Internet Information Services (IIS) server operations errors, request response times, as well as visitors and traffic insights. The app consists of a predefined parser, which provides visibility into your environment for real time or historical analysis.

Log Types

The Sumo Logic app for IIS uses IIS 7.5 logs, which assume the following format:

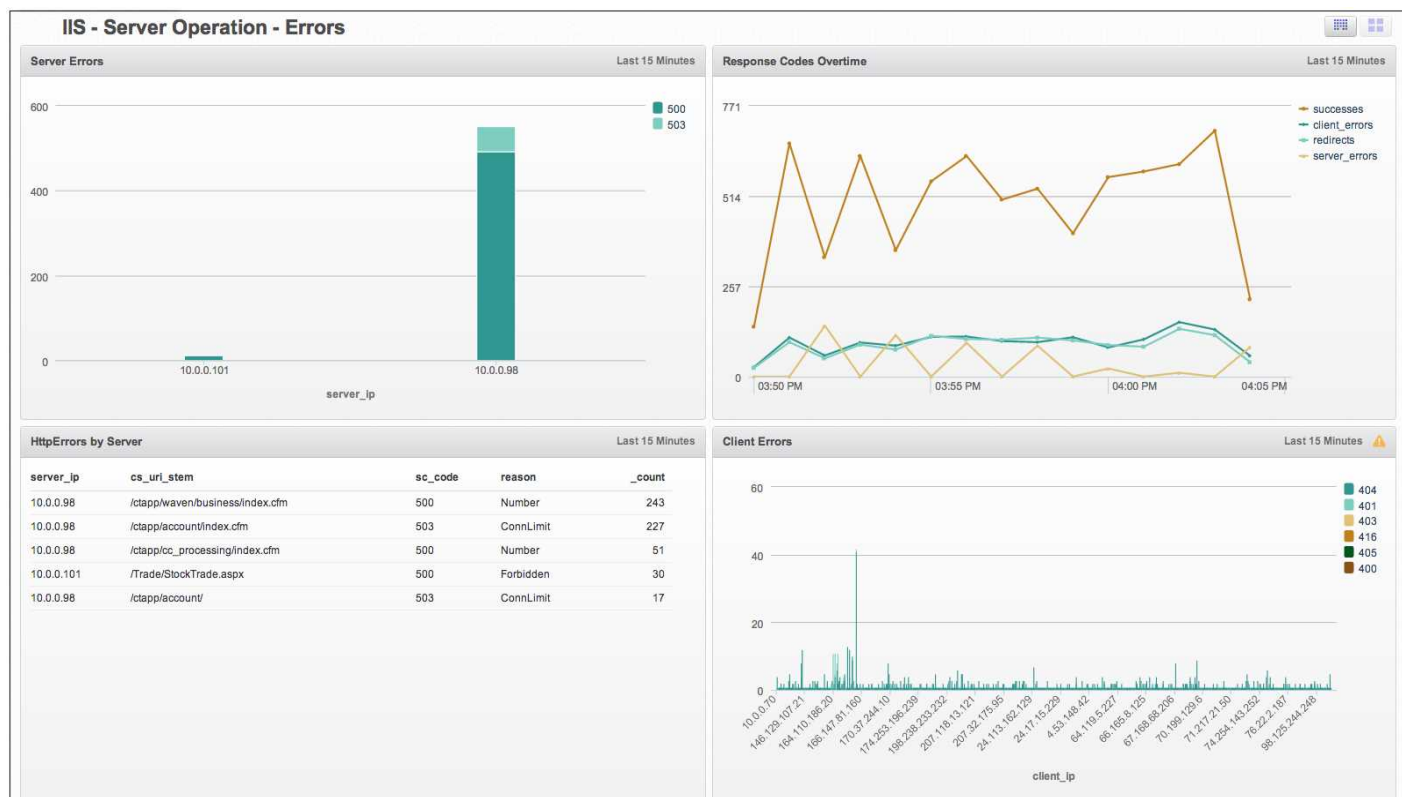
**#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port
cs-username c-ip cs(User-Agent) sc-status sc-substatus
sc-win32-status time-taken**

For details on setting fields to log, see [http://technet.microsoft.com/en-us/library/cc754702\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc754702(v=ws.10).aspx).

Sumo Logic App for IIS Dashboards

The Sumo Logic application for IIS consists of five Dashboards that give you instant access to your system overview, including errors, requests and response time, traffic insights, and visitor insights.

Server Operation - Errors



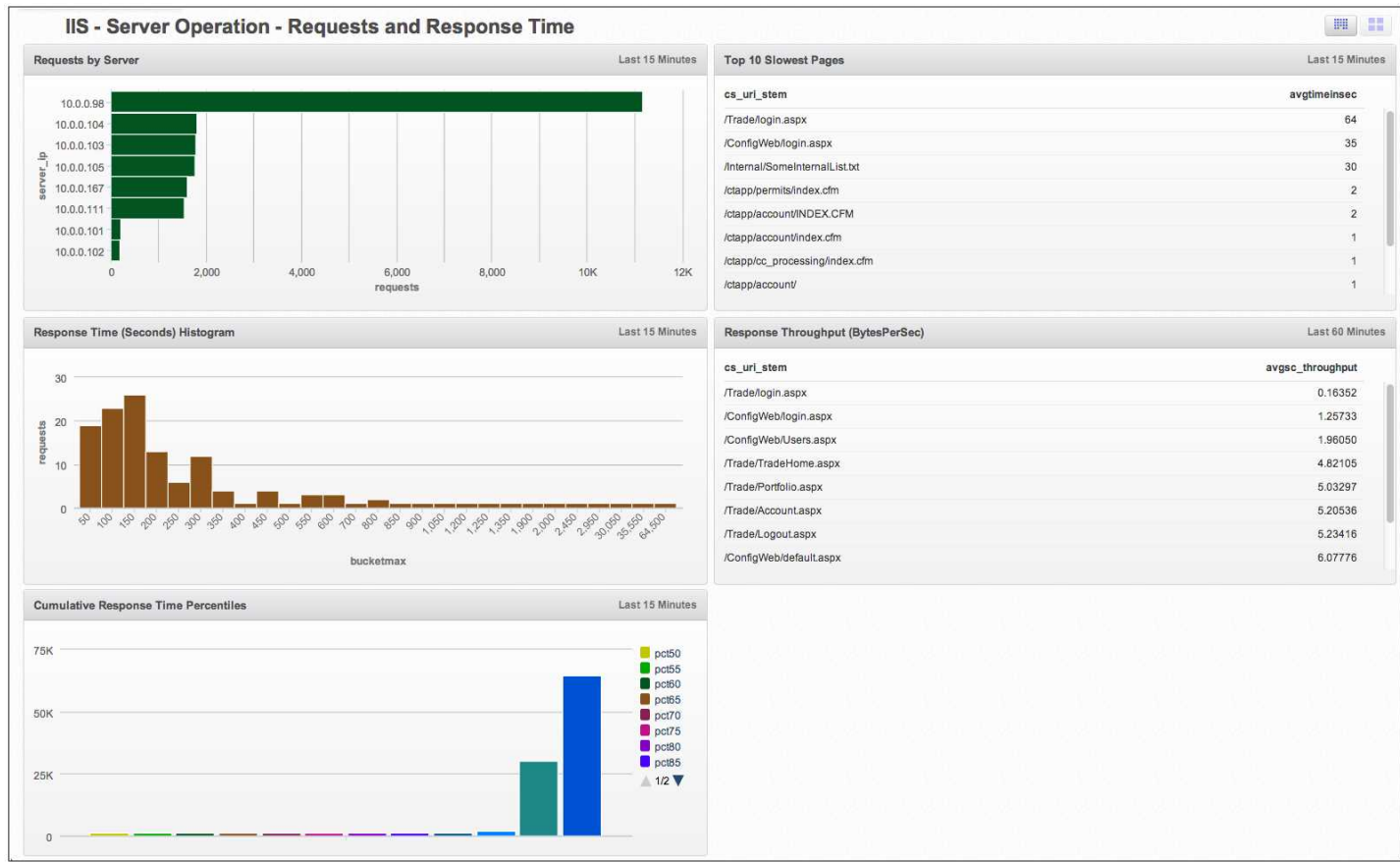
Server Errors. Displays and identifies server error numbers by server IP address for the last 15 minutes in a stacked column chart.

Response Codes Over Time. Provides data on response codes for successes, client errors, redirects, and server errors for the last 15 minutes in an easy to read time line graph. Data is provided in one minute time slices. For more details, click a line to open the graph in the Sumo Logic Web Application **Search** tab.

HTTP Errors by Server. Shows the top five HTTP errors by server IP address for the last 15 minutes in an aggregation table. Includes information on the collector source URI stem, the error code, the reason, and the count of errors.

Client Errors. Displays client errors by client IP address and by number for the last 15 minutes in a column chart.

Server Operation - Requests and Response Time Dashboard



Requests by Server. Displays the number of requests by server IP address for the last 15 minutes in a bar chart.

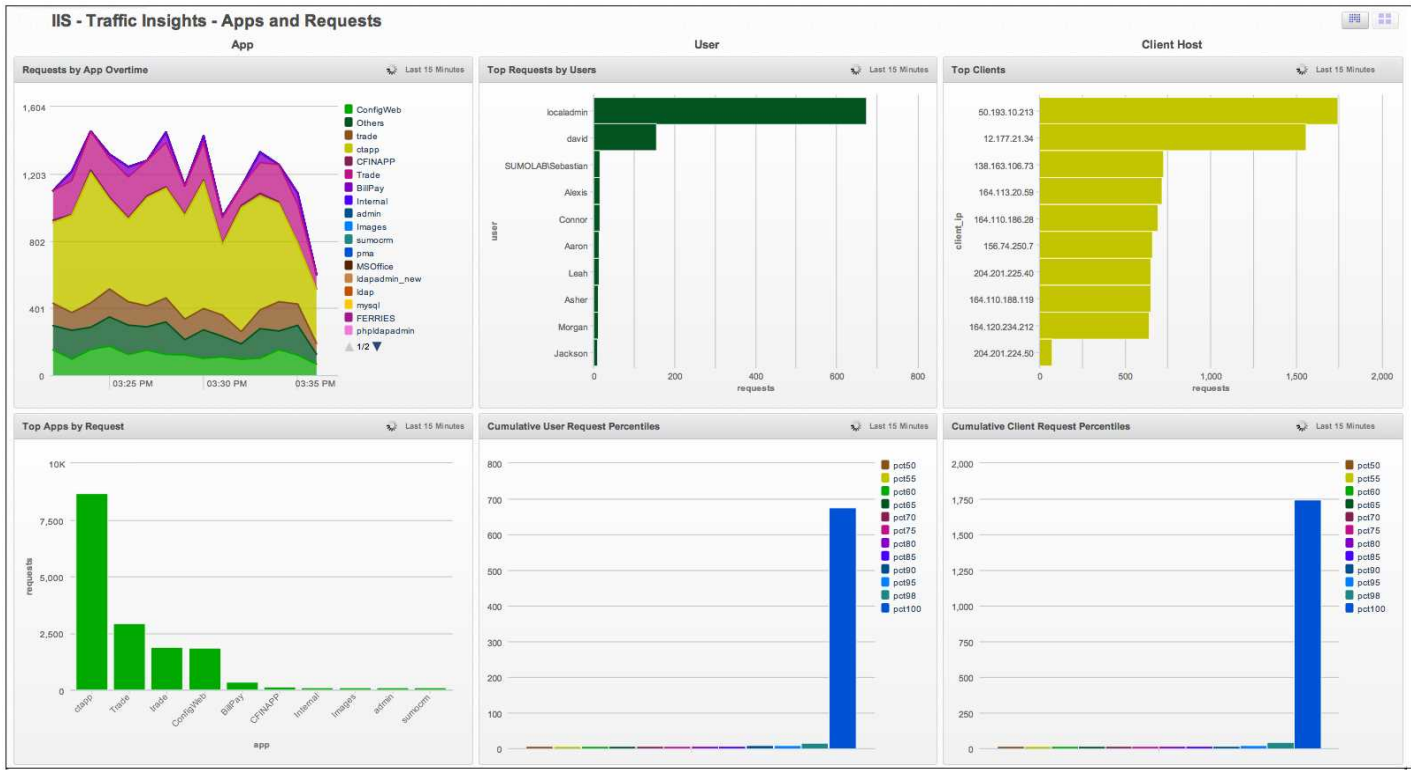
Top 10 Slowest Pages. Provides an aggregate table of the top 10 slowest web pages for the last 15 minutes, including the collector source URI stem, and the average time it takes to respond in seconds.

Response Time (in Seconds). Shows the number of requests and the response time per timeslice for the last 15 minutes in a histogram.

Response Throughput (Bytes per Second). Displays an aggregate table of the top 10 server IP addresses response throughput time in bytes per second for the last hour. Includes information on the collector source URI stem and the average throughput.

Cumulative Response Time Percentiles. Provides the cumulative response times percentiles for the last 15 minutes in a color coded column chart.

Traffic Insights - Apps and Requests



Requests by App Over Time. Counts the number of requests by application in one minute time slices for the last 15 minutes. Displays the data in a stacked area chart.

Top Apps by Request. Displays the names of the top 10 applications and the number of requests received for the last 15 minutes. Data is displayed in a column chart.

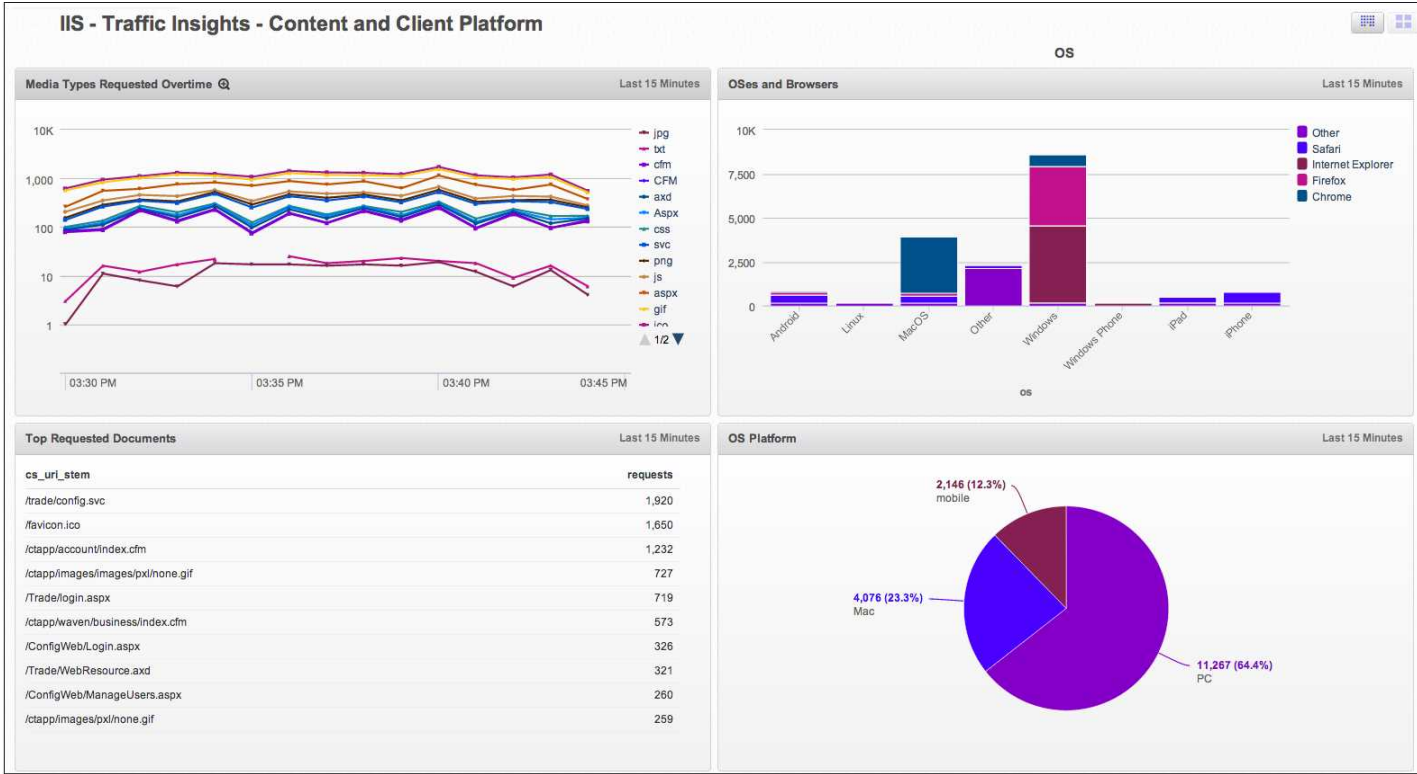
Top Requests by Users. Shows the top 10 users making requests of the system, and the number of requests they have made for the last 15 minutes. Data is provided in a bar chart.

Cumulative User Request Percentiles. Provides the cumulative user request percentiles for the last 15 minutes. Data is displayed as a column chart.

Top Clients. Shows the top 10 clients by IP address and the number of requests they have made for the last 15 minutes. Data is displayed as a bar chart.

Cumulative Client Request Percentiles. Displays the cumulative client IP address request percentiles for the last 15 minutes. Data is displayed as a column chart.

Traffic Insights - Content and Client Platform



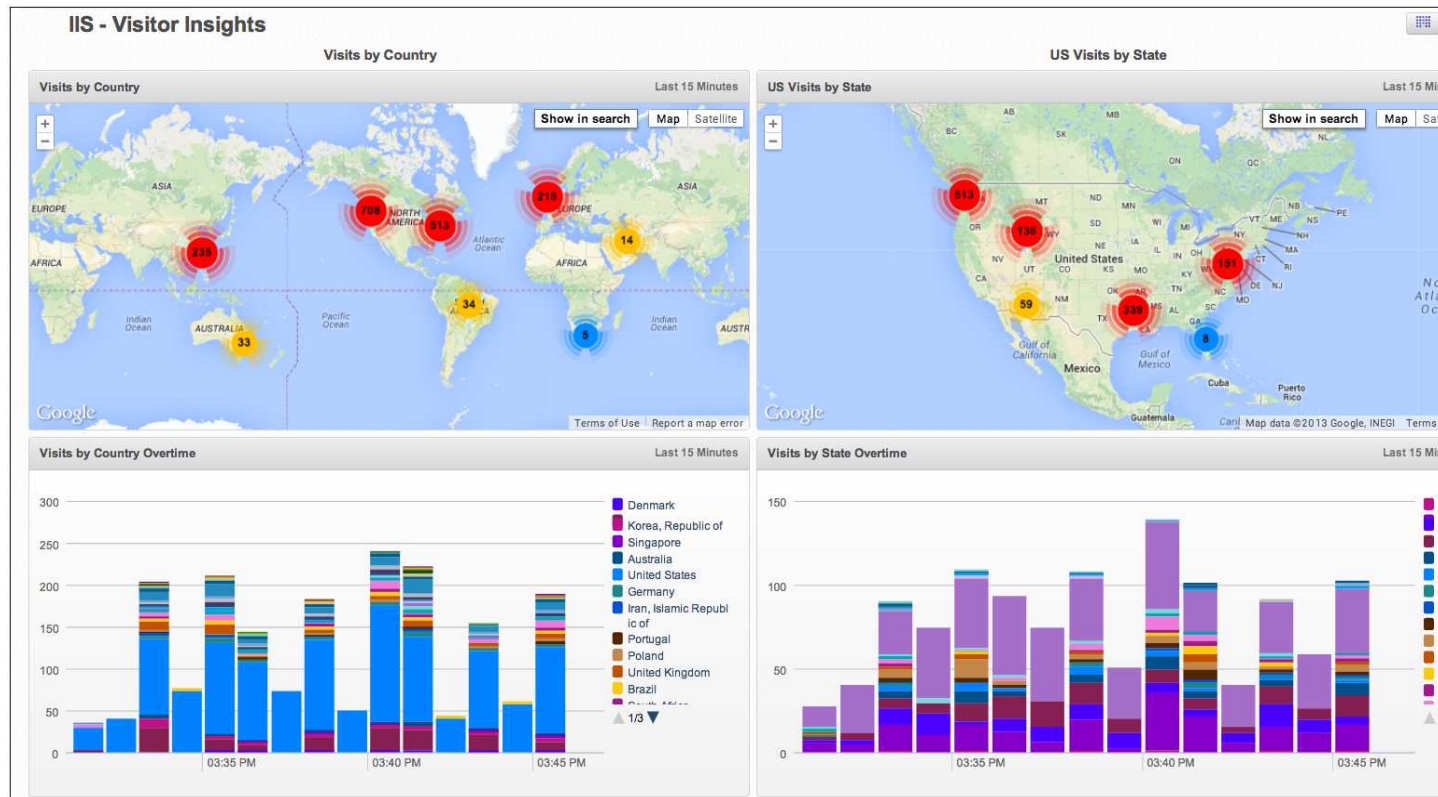
Media Types Requested Overtime. Displays the number of media file types requested by one minute time slices over the last 15 minutes. Data is displayed in a color coordinated time line.

Top Requested Documents. Shows a table listing the top 10 documents requested in the last 15 minutes. Information includes the collector source URI stem and the number of requests.

OSes and Browsers. Allows you to view the operating systems and web browsers used by visitors in the last 15 minutes, displayed as a color coordinated stacked column chart.

OS Platform. Displays the operating system platforms used by visitors over the last 15 minutes, shown in a pie chart.

Visitor Insights



Visits by Country. Performs a geo lookup query to display the locations and number of users by client IP address for the last 15 minutes on a map of the world. Click **Show in search** to display the data in the Web Application Search tab.

Visits by Country Over Time. Displays the number of visitors per country in the last 15 minutes in a stacked column chart time line. Information is provided in time slices of one minute.

US Visits by State. Performs a geo lookup query to display the locations and number of users by client IP address by state on a map of the US. Information is provided for the last 15 minutes. Click **Show in search** to display the data in the Web Application Search tab.

Visits by State Over Time. Displays the number of visitors per US state in the last 15 minutes in a stacked column chart time line. Information is provided in time slices of one minute.

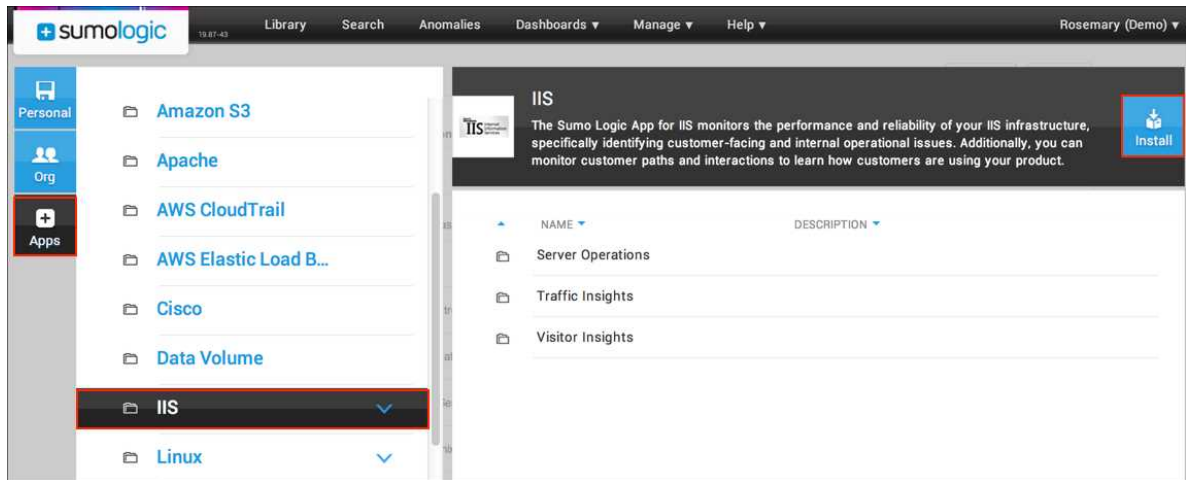
Installing the IIS App

The **Library** makes it easy to install the Sumo Logic Applications your organization needs.

To install the app:

1. In the **Library**, click the **Apps** tab.
2. Click **IIS**.

3. Click **Install**.

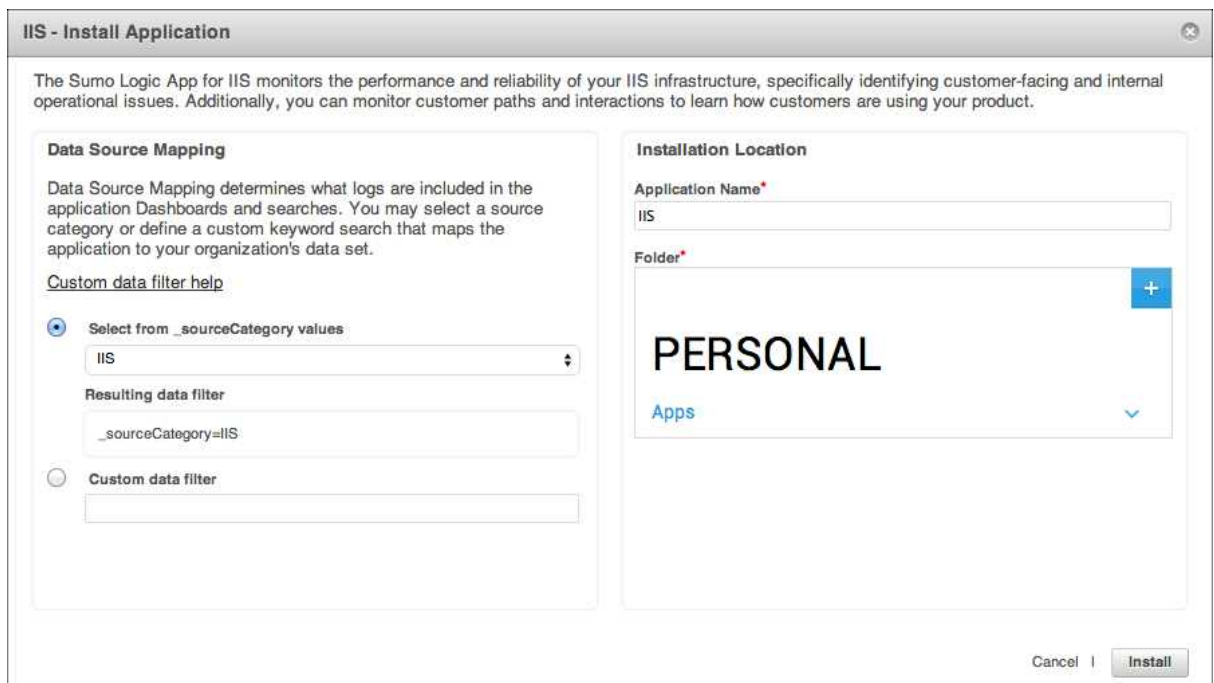


4. In the **Install Application** dialog box, do one of the following for **Data Source Mapping**:

- **Select from _sourceCategory values.** Choose an existing _sourceCategory present in your account used for IIS, such as **IIS**.

Important: If you do not select the correct _sourceCategory, data will not be loaded into the app. If you don't know which _sourceCategory to select, ask the administrator who configured the Source.

- **Custom data filter.** To set up a specific data filter, type the keyword(s) you'd like to use to filter the data.



5. For **Folder**, choose either the **Personal** folder or a subfolder in the **Personal** folder. (Click the blue + to create a new subfolder).
6. Click **Install**.

Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

Sumo Logic App for Linux

The Sumo Logic application for Linux allows you to view the events, logins, and security status of your Linux system. The app consists of predefined searches and three Dashboards, which provide visibility into your environment for real time or historical analysis.

Log Types

Sumo Logic apps gather data from the log messages collected from Sources by Collectors. The Sumo Logic app for Linux requires the following log types, which are set up during the Collector and Source configuration process:

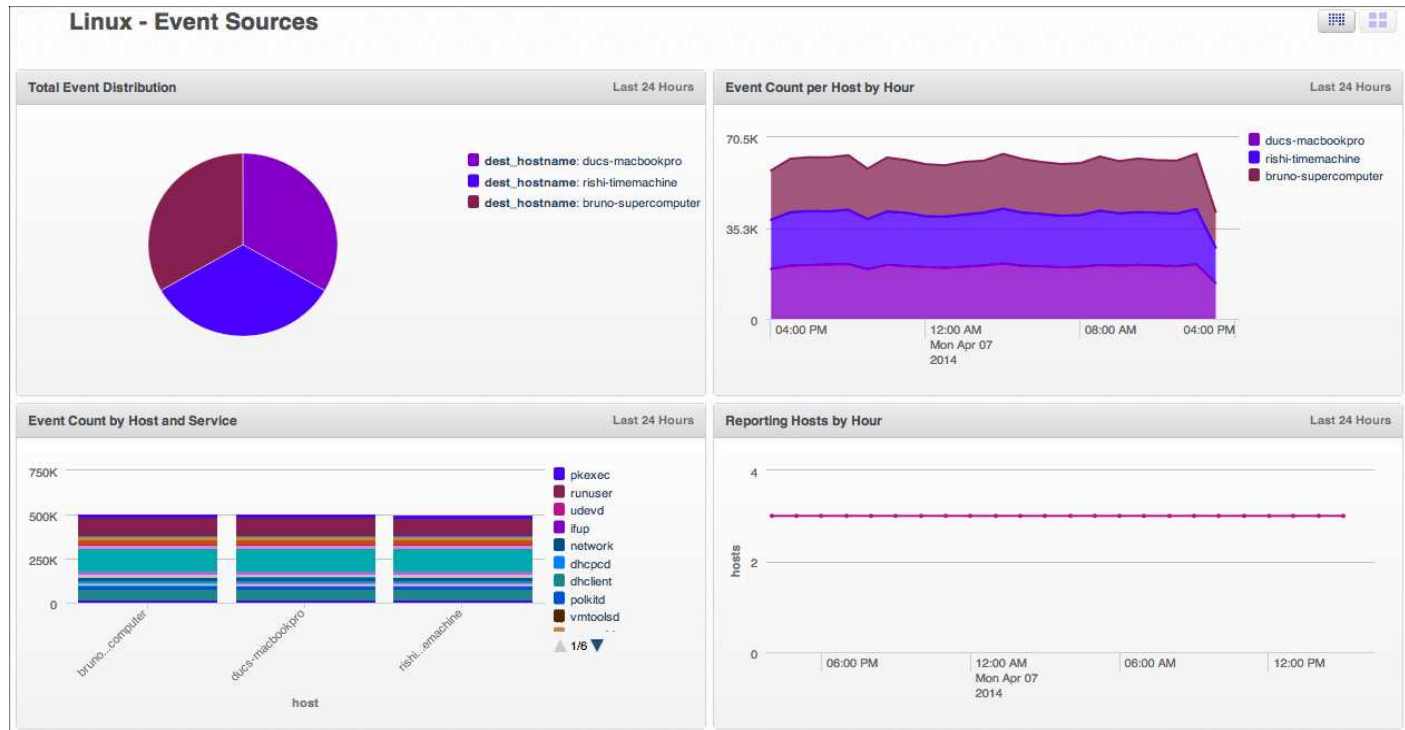
- **/var/log/messages*** - These logs contain system messages. They are required for most system events, such as user creation, deletion, system start, shutdown, etc.
- **/var/log/audit*** or **/var/log/secure*** - The log type used will depend on the version of UNIX and configuration. These logs contain security logs. They are required for most security events and user logins.
- **/var/log/ [yum.log | dpkg.log | zypper.log]** - Optional: These logs are required for package operation searches.

It is recommended to categorize all of these logs uniformly with a single source category, for example: OS/Linux.

Sumo Logic App for Linux

The Sumo Logic application for Linux includes three Dashboards that give you instant access to your system overview, including event sources, login status, and security status.

Event Sources



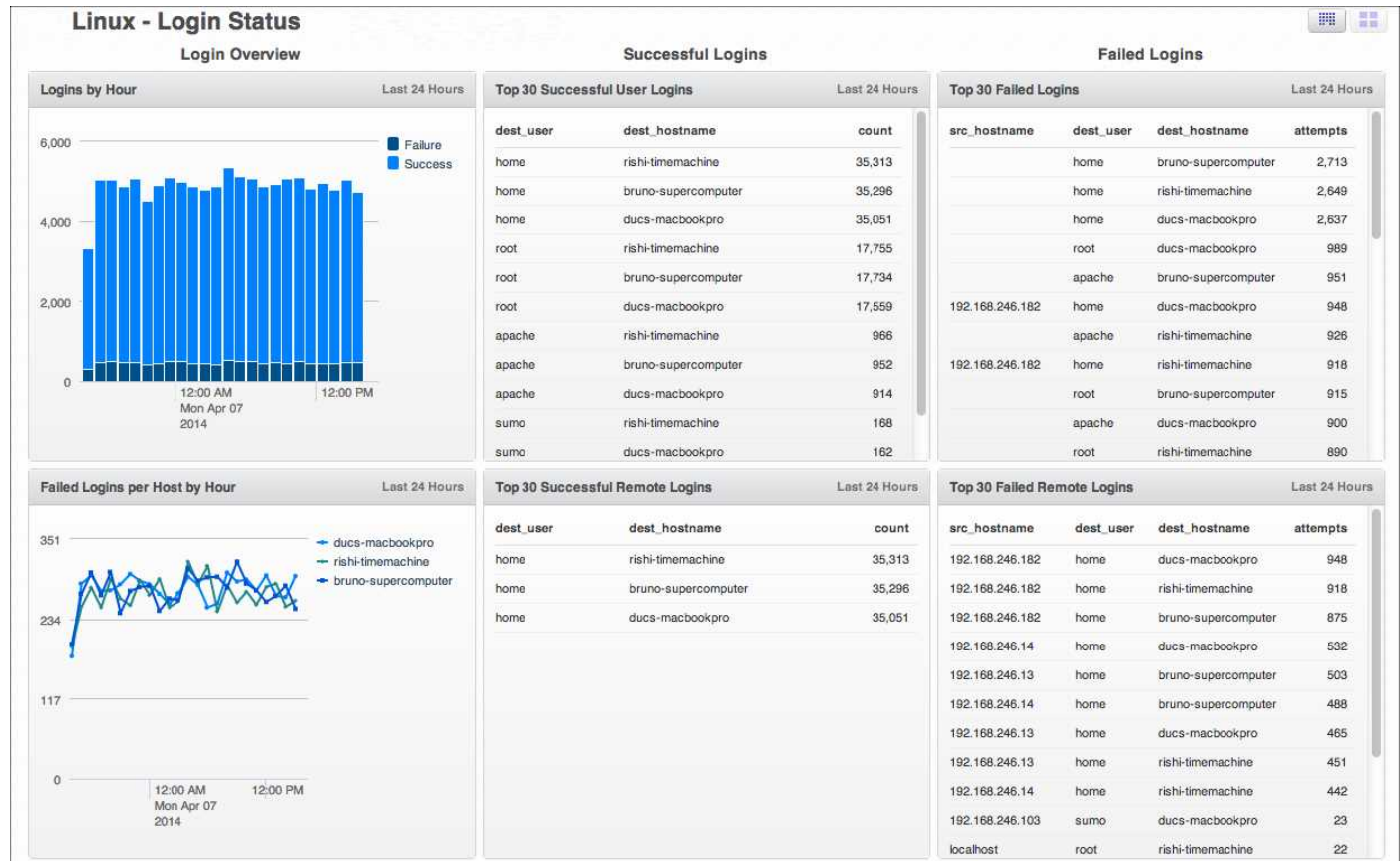
Total Event Distribution. Displays the total number of events by destination host name over the last 24 hours in a pie chart.

Event Count by Host and Service. Shows the total number of events by host name and service name for the last 24 hours, displayed as a stacked column chart.

Event Count per Host by Hour. Provides the number of events per host name by hour for the last 24 hours, displayed as an time line area chart.

Reporting Hosts by Hour. Displays the number of hosts reporting by hour for the last 24 hours in a time line chart.

Login Status



Logins by Hour. Displays the number of user logins by hour over the last 24 hours in a stacked column chart.

Successes and failures are displayed in contrasting colors.

Failed Logins per Host by Hour. Shows the failed user logins per host by hour for the last 24 hours in a time line chart, which allows you to easily identify any login problems immediately.

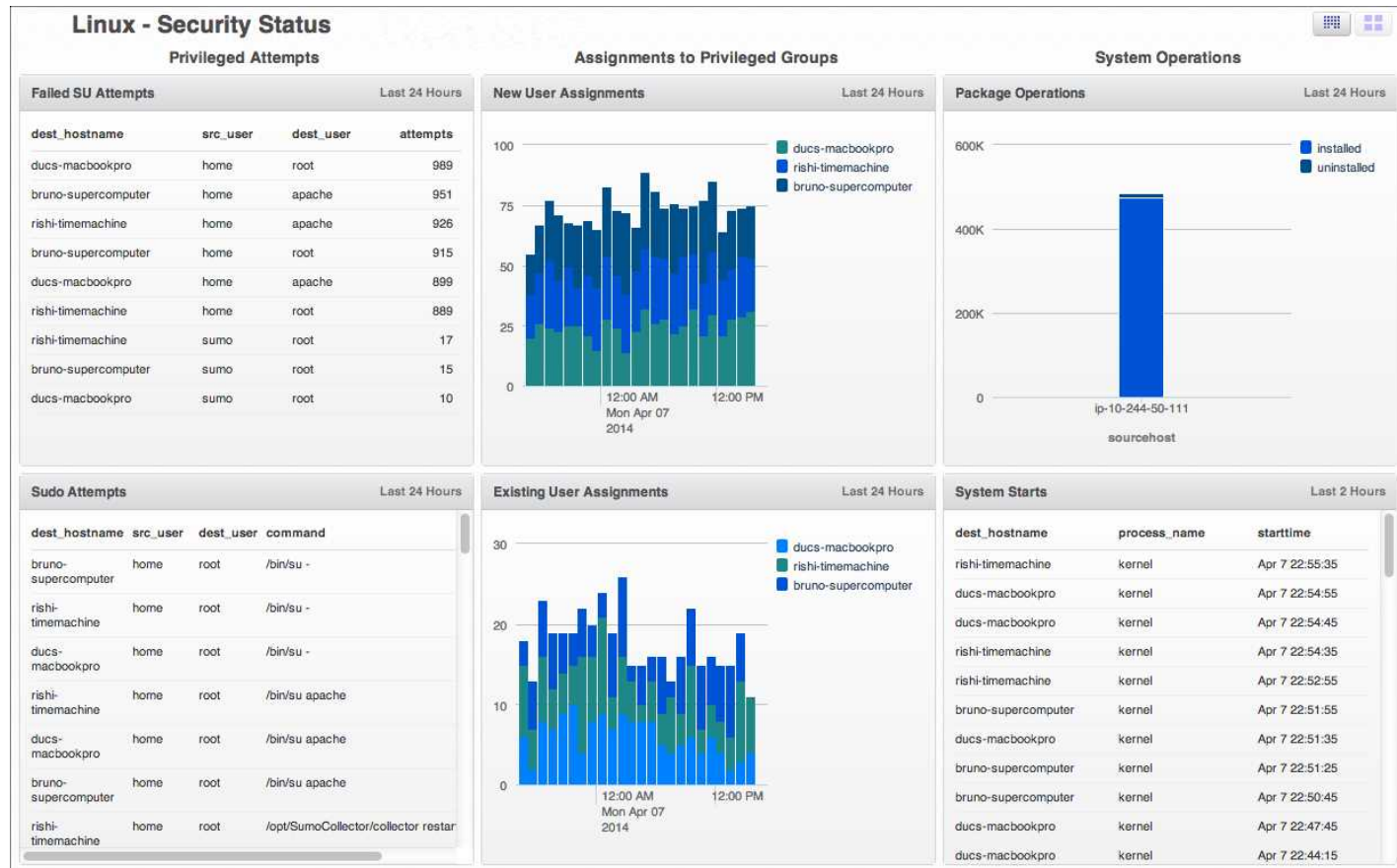
Top 30 Successful User Logins. Provides an aggregation table that displays the top 30 successful user logins for the last 24 hours. Information includes the user, the hostname, and the login count.

Top 30 Successful Remote Logins. Shows an aggregation table of the top 30 successful remote logins for the last 24 hours. Information includes the user, the hostname, and the login count.

Top 30 Failed Logins. Displays an aggregation table that details the top 30 failed logins over the last 24 hours. Information includes the source hostname, user, destination hostname, and number of attempts.

Top 30 Failed Remote Logins. Provides an aggregation table of the top 30 failed remote login attempts over the last 24 hours. Information includes the source hostname, user, destination hostname, and number of attempts.

Security Status



Failed SU Attempts. Displays an aggregation table that details failed SU (superuser) attempts for the last 24 hours. Information includes the destination hostname, source user, destination user, and the number of attempts.

Sudo Attempts. Shows an aggregation table that provides information on Sudo attempts for the last 24 hours. Information includes the destination hostname, source user, destination user, command, and the number of attempts.

New User Assignments. Provides information on the number of new user assignments by host by hour for the last 24 hours, displayed in a stacked column chart.

Existing User Assignments. Displays the number of existing user assignments by host by hour for the last 24 hours, displayed in a stacked column chart.

Package Operations. Shows the number of package operations, both installed and uninstalled, performed on a source host for the last 24 hours in a stacked column chart.

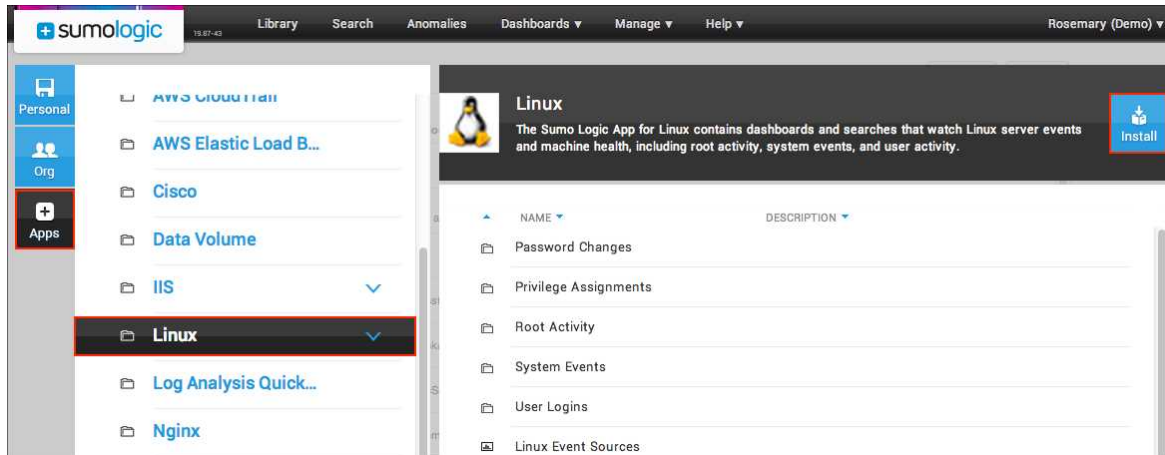
System Starts. Provides an aggregation table with information on system starts for the last two hours. Information includes the destination hostname, the process name, and the start time.

Installing the Linux App

The **Library** makes it easy to install the Sumo Logic Applications your organization needs.

To install the app:

1. In the **Library**, click the **Apps** tab.
2. Click **Linux**.
3. Click **Install**.



4. In the **Install Application** dialog box, do one of the following for **Data Source Mapping**:

A screenshot of the 'Linux - Install Application' dialog box. The dialog has a title bar with a close button. Below the title bar is a description: 'The Sumo Logic App for Linux contains dashboards and searches that watch Linux server events and machine health, including root activity, system events, and user activity.' The dialog is divided into two main sections. The left section is titled 'Data Source Mapping' and contains a text box explaining that data source mapping determines what logs are included. Below this is a dropdown menu labeled 'Select from _sourceCategory values' with 'OS/Linux' selected. Below the dropdown is a text box labeled 'Resulting data filter' containing the text '_sourceCategory=OS/Linux'. There is also an option for 'Custom data filter' which is currently unselected. The right section is titled 'Installation Location' and contains a text box for 'Application Name' with 'Linux' entered. Below this is a section for 'Folder' which is currently empty. At the bottom right of the dialog are 'Cancel' and 'Install' buttons.

- **Select from _sourceCategory values.** Choose an existing _sourceCategory present in your account used for Linux, such as **OS/Linux**.
Important: If you do not select the correct _sourceCategory, data will not be loaded into the app. If you don't

know which `_sourceCategory` to select, ask the administrator who configured the Source.

Linux - Install Application

The Sumo Logic App for Linux contains dashboards and searches that watch Linux server events and machine health, including root activity, system events, and user activity.

Data Source Mapping

Data Source Mapping determines what logs are included in the application Dashboards and searches. You may select a source category or define a custom keyword search that maps the application to your organization's data set.

[Custom data filter help](#)

☐ Select from `_sourceCategory` values

Select a Category...

Resulting data filter

☐ Custom data filter

`_sourceCategory=OS/Linux AND _collector=prod*`

Installation Location

Application Name*

Linux

Folder*

PERSONAL

Data Volume

Windows

Cancel | Install

- **Custom data filter.** To set up a specific data filter, type the search expression you'd like to use to filter the data. For example, if you want this app to be used only with production collectors that begin with the word "prod", select a custom data filter such as `_sourceCategory=OS/Linux AND _collector=prod*`.
5. For **Folder**, choose either the **Personal** folder or a subfolder in the **Personal** folder. (Click the blue + to create a new subfolder).
 6. Click **Install**.

Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

Sumo Logic App for Nginx

The Sumo Logic App for Nginx provides searches and Dashboards that keep an eye on log events generated by Nginx servers. Events in Dashboards are divided into the following categories:

- **Deployment overview.** Get an overall look at the activity of the sites running on Nginx servers.
- **Visitor locations.** Know at a glance where your visitors originate.
- **Visitor access types.** Gather insights on the devices and operating systems visitors are using to access your sites.
- **Visitor traffic information.** Find out which external sites are referring your visitors. Additionally, you can quickly view the amount and types of media being served up to visitors.
- **Web server ops.** Learn more about the errors generated from your Nginx servers.

Log Types

The Sumo Logic App for Nginx assumes the NCSA extended/combined log file format for Access logs and the default Nginx error log file format for error logs.

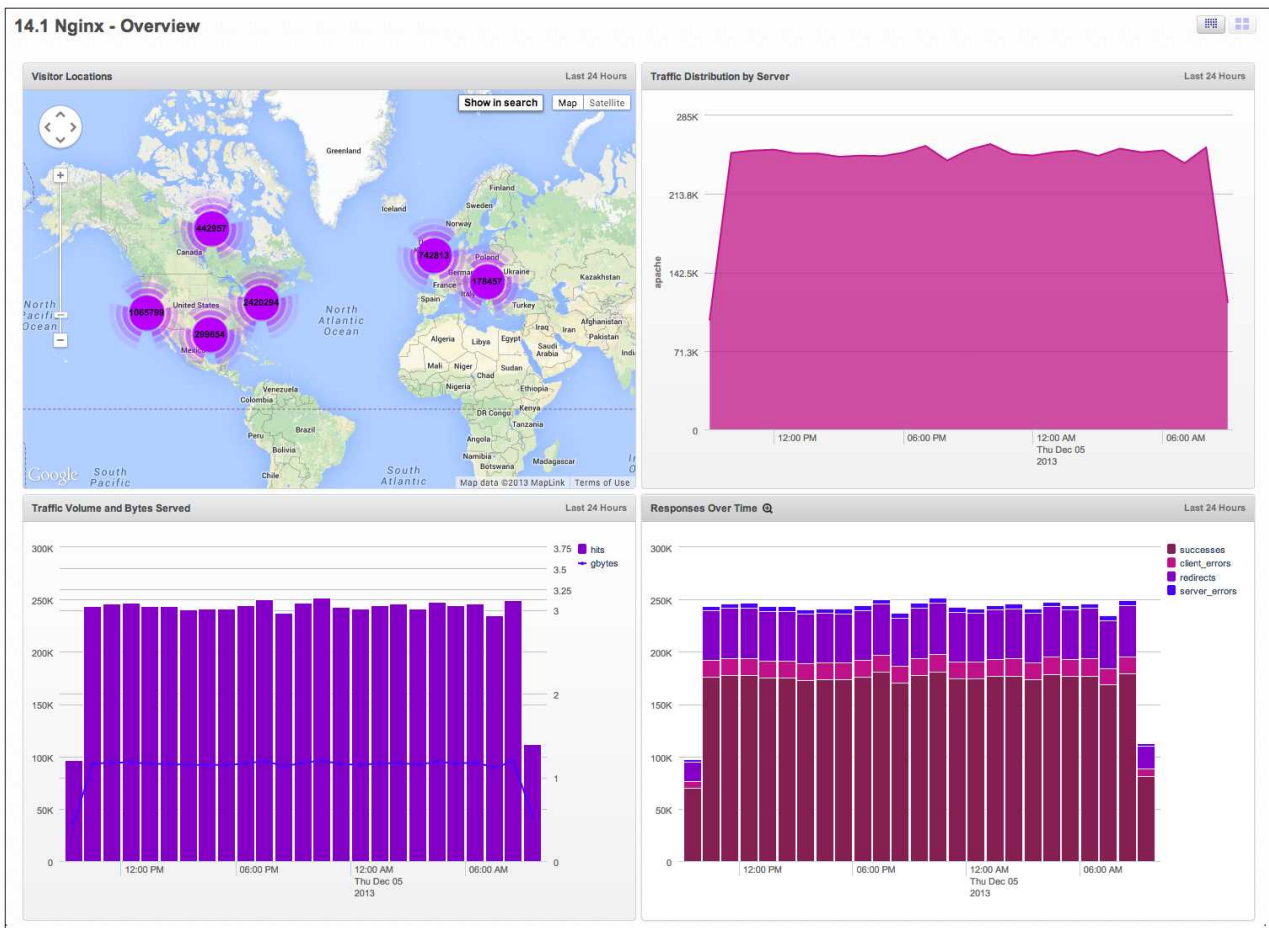
All Dashboards (except the Web Server Operations dashboard) assume the Access log format. The Web Server Operations Dashboard assumes both Access and Error log formats, so as to correlate information between the two.

For more details on the Access log file format, see http://httpd.apache.org/docs/current/mod/mod_log_config.html.

Sumo Logic App for Nginx Dashboards

Nginx Overview Dashboard

This dashboard shows Monitors that give you an overall look at activity of your site(s).



Visitor Locations. This geolocation Monitor displays global visitors to your site(s).

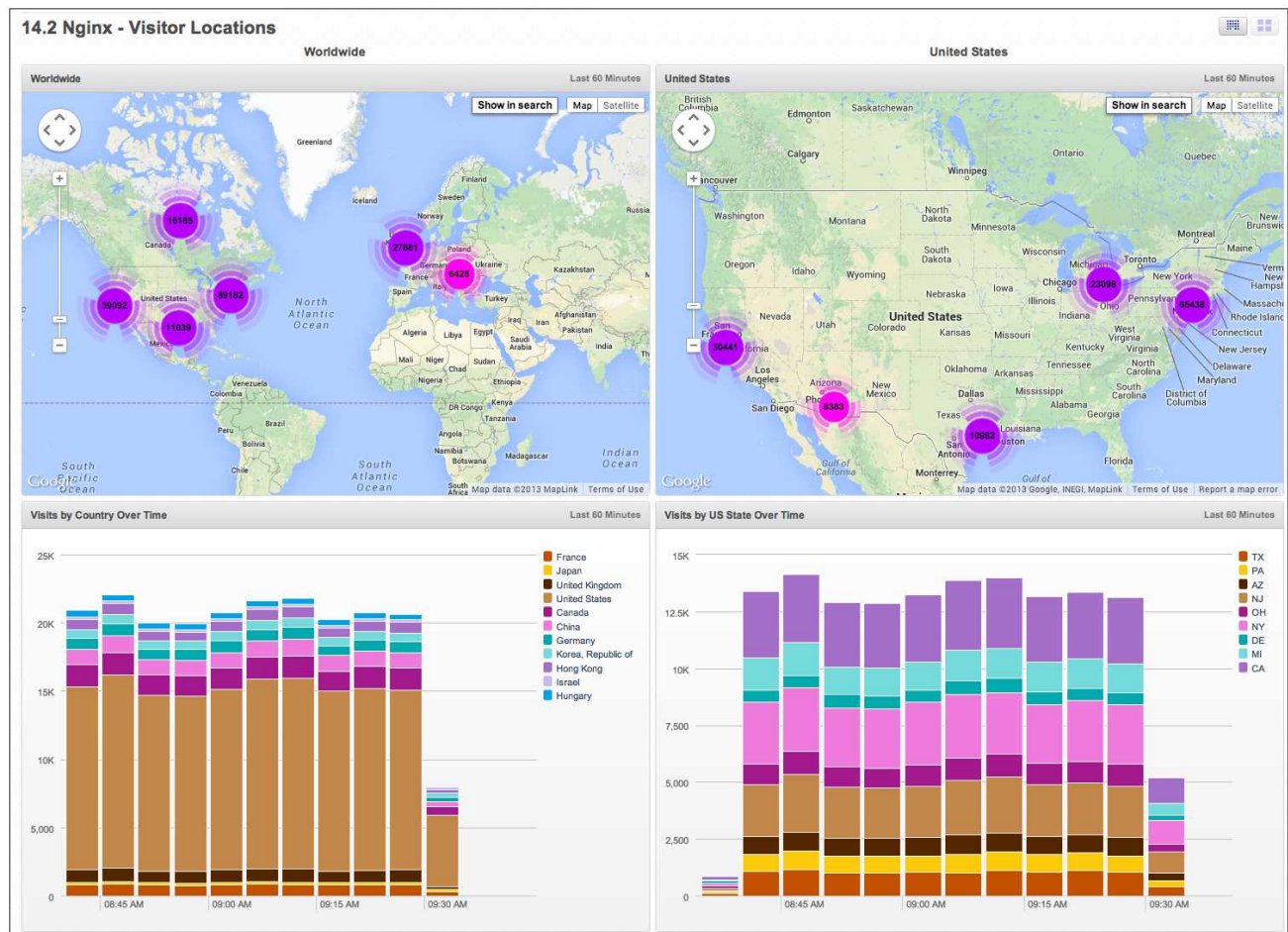
Traffic Distribution by Server. Traffic to each server is displayed, allowing you to see any unexpected changes in volume.

Traffic Volume and Bytes Served. This combo chart displays the gigabytes served as a line over the number of hits your site received.

Responses over Time. This Monitor shows response successes, client errors, redirect, and server errors. Spikes in errors or redirects are easy to spot.

Nginx Visitor Locations Dashboard

The Visitor Locations Dashboard displays Monitors that constantly update information about visitors to your sites, both domestic and international.



Worldwide. This geolocation Monitor displays global visitors to your site(s). You can zoom in to inspect specific areas.

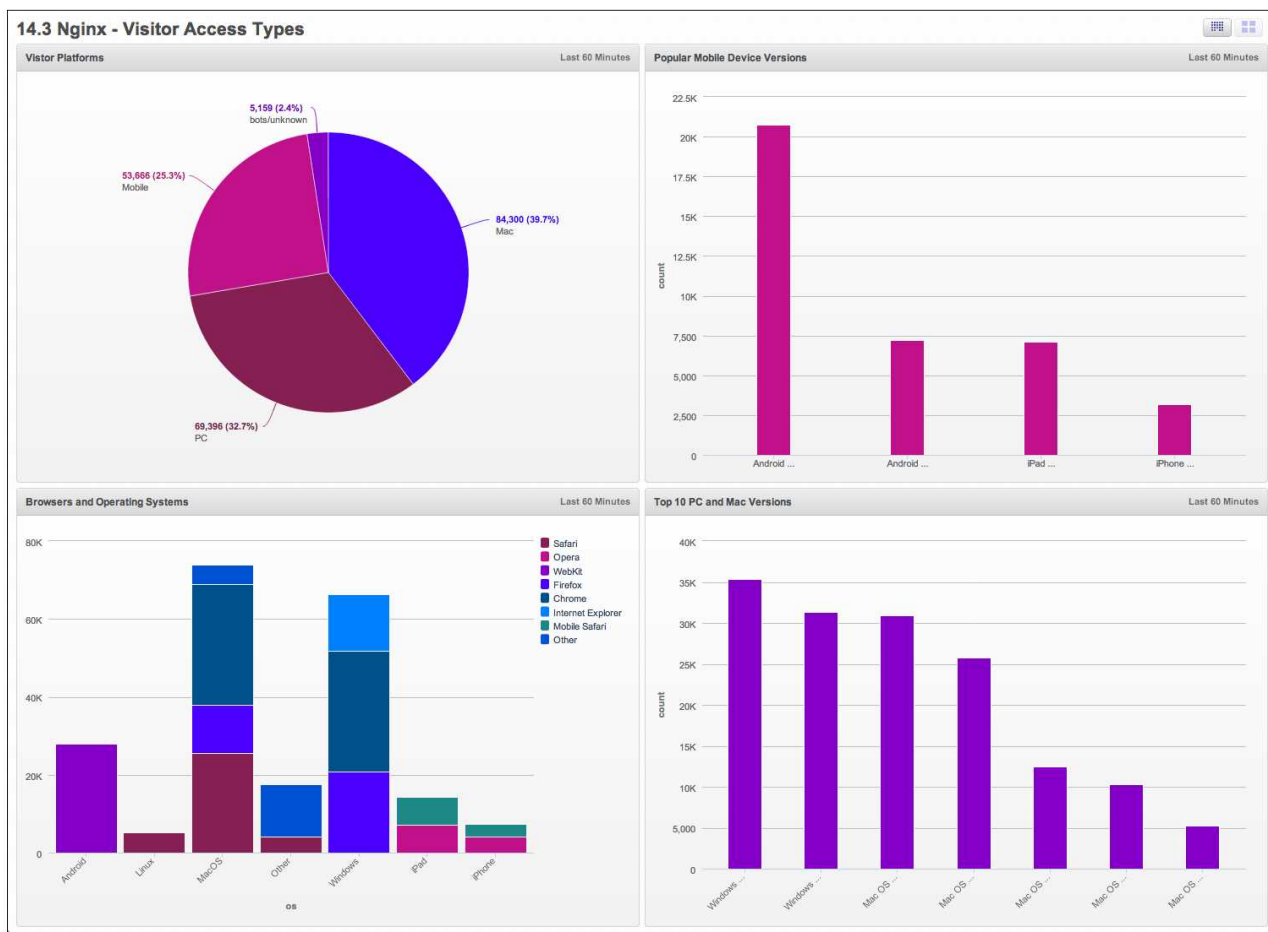
Visits by Country Over Time. Using a query similar to the Visits by US State Monitor, each country is represented in a band in a column chart. Any activity that originates in a suspicious location will be easy to see.

United States. This geolocation Monitor displays visitors from just the US.

Visits by US State Over Time. Uses a query that includes a geo lookup operator (as well as the transpose operator) to display a column chart that represents the disposition of visitors per state.

Nginx Visitor Access Types Dashboard

This Dashboard displays information about the devices and OS versions visitors are using to access your sites.



Visitor Platforms. Breaks down the percentages of users accessing your site on Mac, PC, Mobile, and unknown platforms.

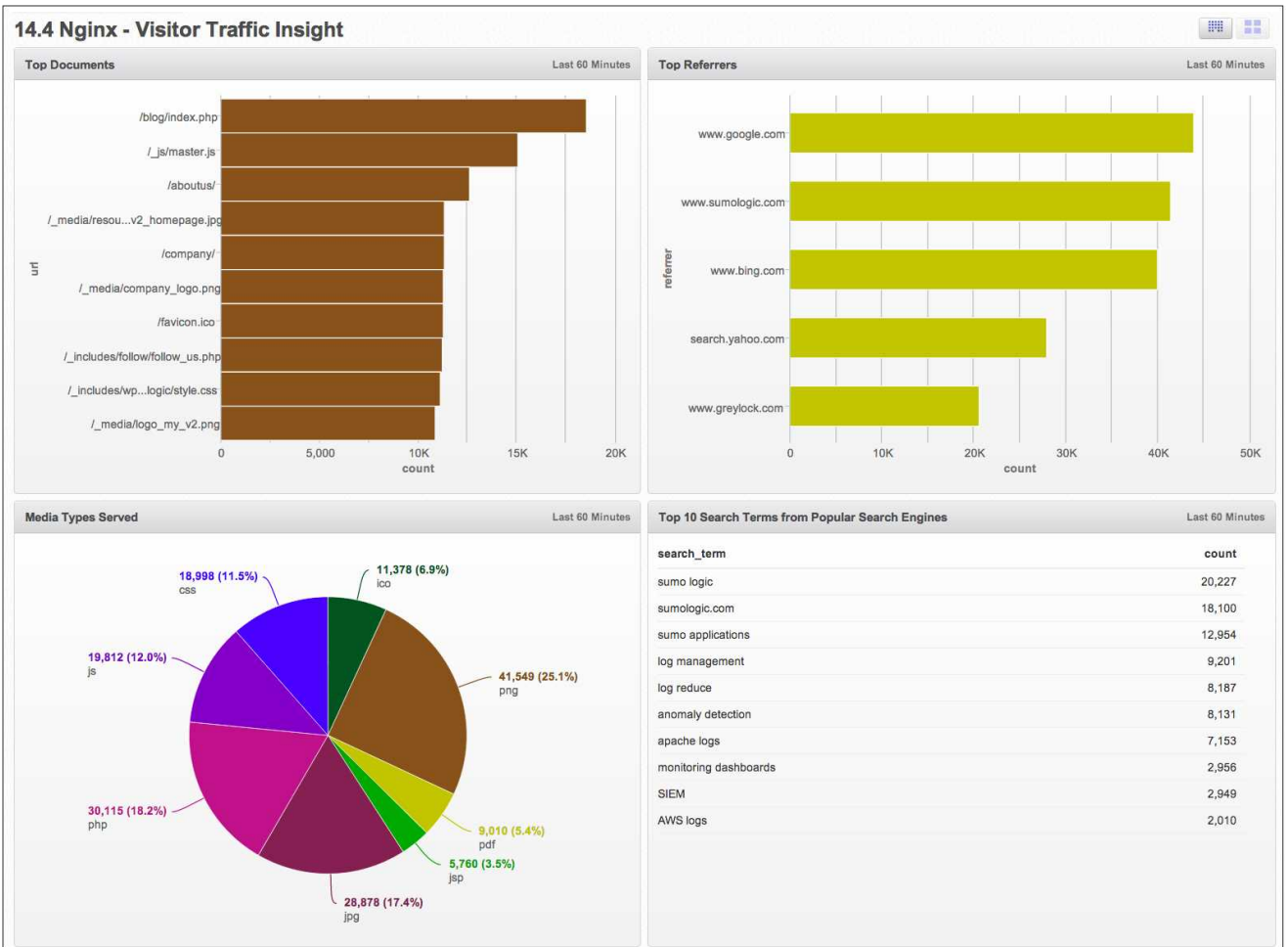
Popular Mobile Device Versions. Breaks down the type of mobile devices accessing your site by Android and iOS (iPhone and iPad users are reported separately).

Top 10 PC and Mac Versions. Need to understand which versions of operating systems your users have installed? This Monitor displays each version of Mac and Windows OS installed on visitors' machines, ranked in order of popularity.

Browsers and Operating Systems. Uses a query that parses out both the OS (mobile or computer) and browsers that are hitting your site.

Nginx Visitor Traffic Insight

This Dashboard provides information about the content being served to customers, as well as sites that are referring the most visitors.



Top Documents. Displays the most frequently served content to visitors, including graphics on your site.

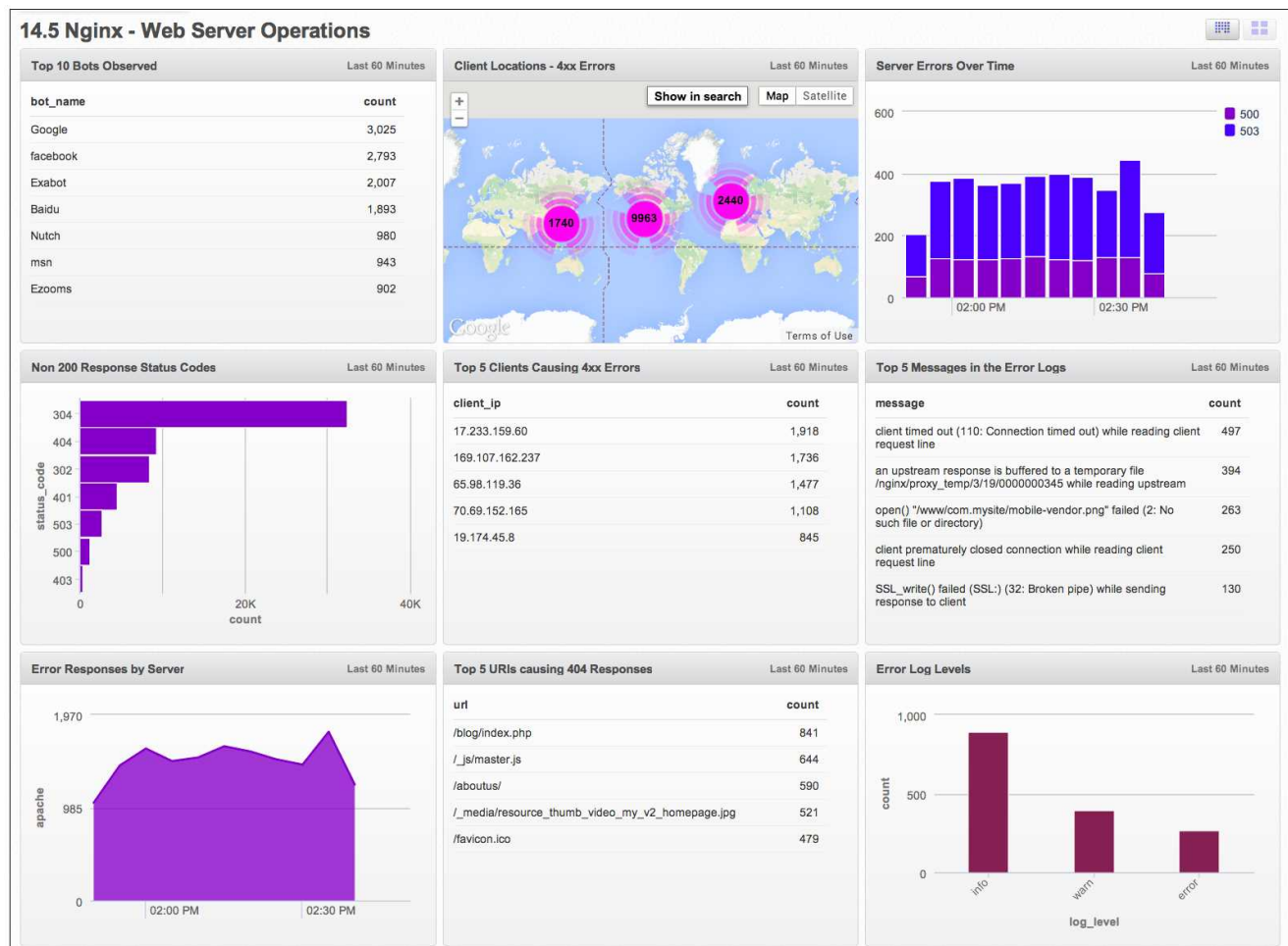
Top Referrers. Wonder where your visitors are coming from? This Monitor shows the top five referring sites.

Media Types Served. No need to guess what the most popular media types are—you'll find them here.

Top 10 Search Terms from Popular Search Engines. This query looks at the search terms that visitors used to find your site. These results can confirm what you may have assumed users were searching for, or can uncover new search terms that you can leverage.

Nginx Web Server Operations

The Monitors in this Dashboard give a deep-dive look at specific corners of your Access logs.



Top 10 Bots Observed. Parses bots from Access logs, searching against commonly-used search engines to find bots.

Client Locations -4xx Errors. A geolocation query displays a map of the IPs from where 4xx errors are originating.

Server Errors Over Time. Keep an eye on the number of server errors (5xx code errors) that occur in chunks of five minutes.

Non-200 Response Status Codes. Displays the number of non-200 responses received, sorted by status or error code. A spike in any particular response code can be immediately viewed and addressed.

Top 5 Clients Causing 4xx Errors. Displays the top five client IP addresses that are responsible for 4xx or client errors.

Top 5 Messages in Error Logs. Displays the five most common error messages, based on the count of errors per message.

Error Responses by Server. Displays errors produced by each server in your deployment,

Top 5 URLs causing 404 Responses. Lists the five URLs that are generating the most 404 errors. By having those URLs handy, any forensic investigation can begin right away.

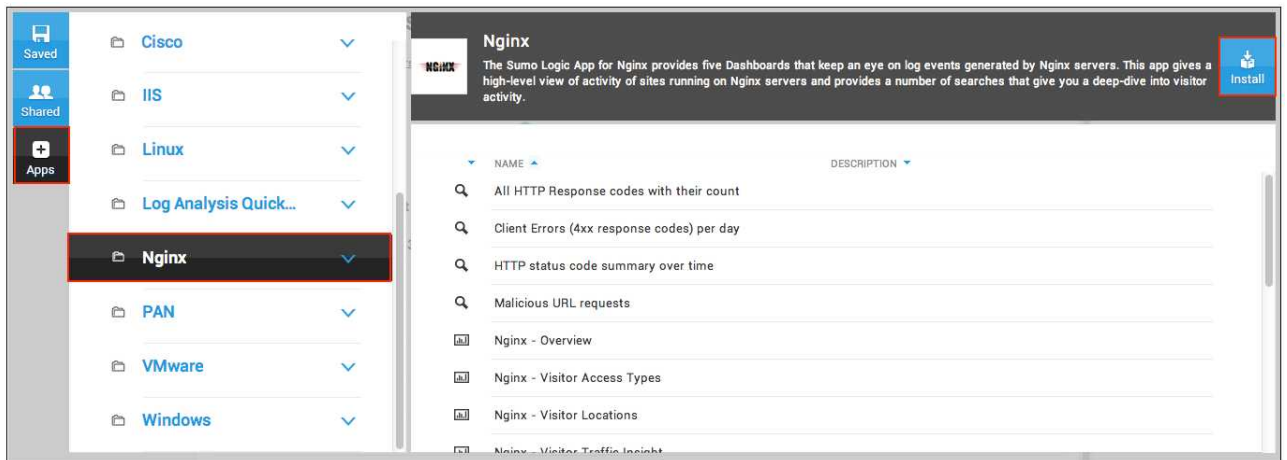
Error Log Levels. Parses out different logging levels present in your data.

Installing the Nginx App

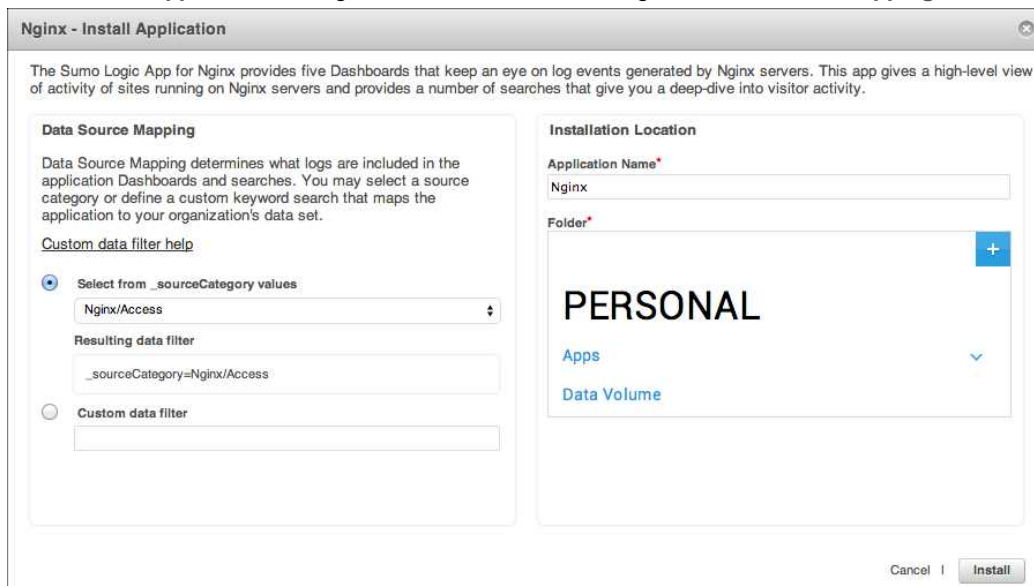
The **Library** makes it easy to install the Sumo Logic Applications your organization needs.

To install the app:

1. In the **Library**, click the **Apps** tab.
2. Click **Nginx**.
3. Click **Install**.



4. In the **Install Application** dialog box, do one of the following for **Data Source Mapping**:



- **Select from _sourceCategory values.** Choose an existing _sourceCategory present in your account used for the Nginx Source, such as **Nginx/Access** or **Apache/Access**.

- To analyze only Nginx Access logs, choose a source category that matches the Nginx Access logs. A majority of searches and Dashboards in this application are written for Nginx Access logs, so some Dashboard monitors and searches that are based on error logs will not work.
 - To monitor only Nginx error logs, choose a source category that matches the Nginx error logs. The majority of searches and Dashboards in this application are written for Nginx Access logs, so most Dashboard monitors and searches that are based on Access logs will not work.
 - **Important:** If you do not select the correct `_sourceCategory`, data will not be loaded into the app. If you don't know which `_sourceCategory` to select, ask the administrator who configured the Source.
 - **Custom data filter.** To set up a specific data filter, type the keyword(s) you'd like to use to filter the data. If you want to analyze both Nginx Access logs and error logs, please use a custom data filter that selects both Access and error log data.
5. For **Folder**, choose either the **Personal** folder or a subfolder in the **Personal** folder. (Click the blue + to create a new subfolder).
 6. Click **Install**.

Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

Sumo Logic App for Palo Alto Networks (PAN)

The Sumo Logic App for Palo Alto Networks is comprised of four distinct Dashboards, giving your organization several different ways to discover threats, consumption, traffic patterns, and other security-driven issues, providing additional insight for investigations. Sumo Logic App for Palo Alto Networks uses Palo Alto Network's threat categories in several Monitors to deliver a graphical representation of threats.

Log Types

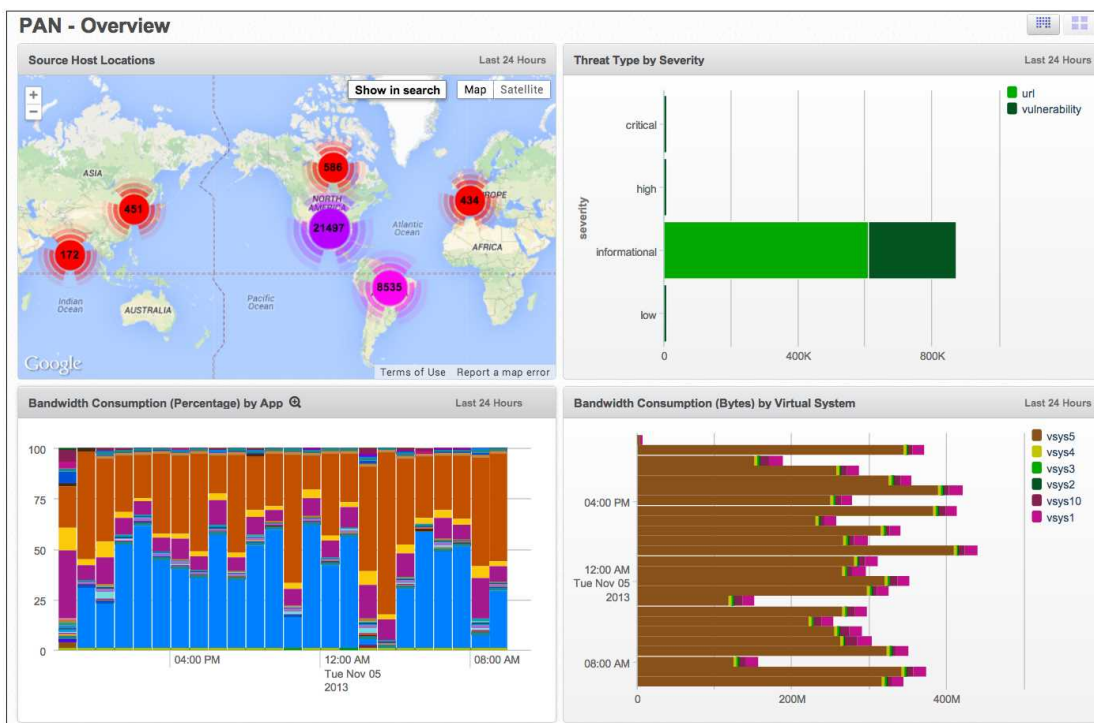
Parsing in the Sumo Logic app for PAN is based on the PAN-OS Syslog integration, which is described in this document:

<https://live.paloaltonetworks.com/servlet/JiveServlet/previewBody/2021-102-8-13361/PANOS-Syslog-Integration-TN-RevM.pdf>

Sumo Logic App for Palo Alto Networks (PAN) Dashboards

Overview Dashboard

The Overview Dashboard keeps you up-to-speed on the higher level operations of your PAN deployment.



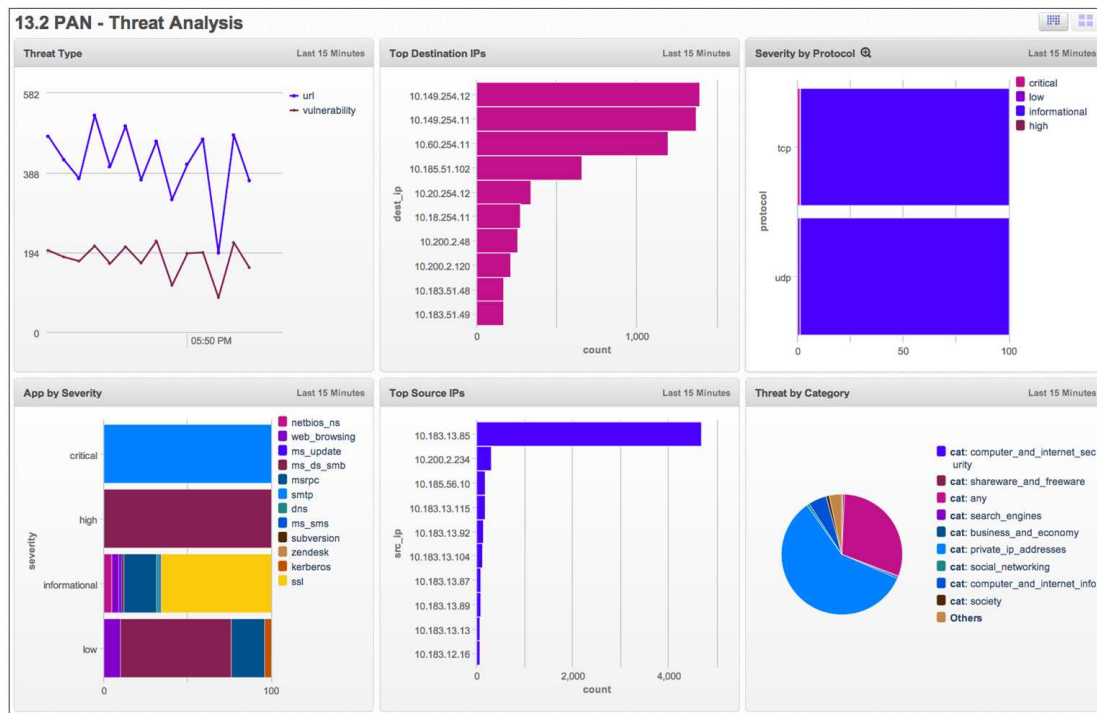
Source Host Locations. Using a geolocation query, this Monitor maps the location of source hosts using their IP addresses.

Threat Type by Severity. Breaks down the number of threats, ranked by severity; threat types are divided into separate categories (such as Vulnerabilities and URL). Threat types displayed in this Monitor include Low, Informational, High, and Critical.

Bandwidth Consumption (Bytes) by Virtual System. Displays the bandwidth of virtual systems, making it easy to see which systems are consuming the most bandwidth.

Bandwidth Consumption (Percentage) by App. Each app deployed by your organization is represented in an overall breakdown of how apps are consuming bandwidth.

Threat Analysis Dashboard



Threat Type. Get an idea of the number of threats as well as the type of threats detected by Palo Alto Networks.

Top Destination IPs. Shows the top 10 destination IPs (the IPs that have made the most attempts).

Severity by Protocol. View the number of threats sorted by severity (Critical, High, Low, or Informational).

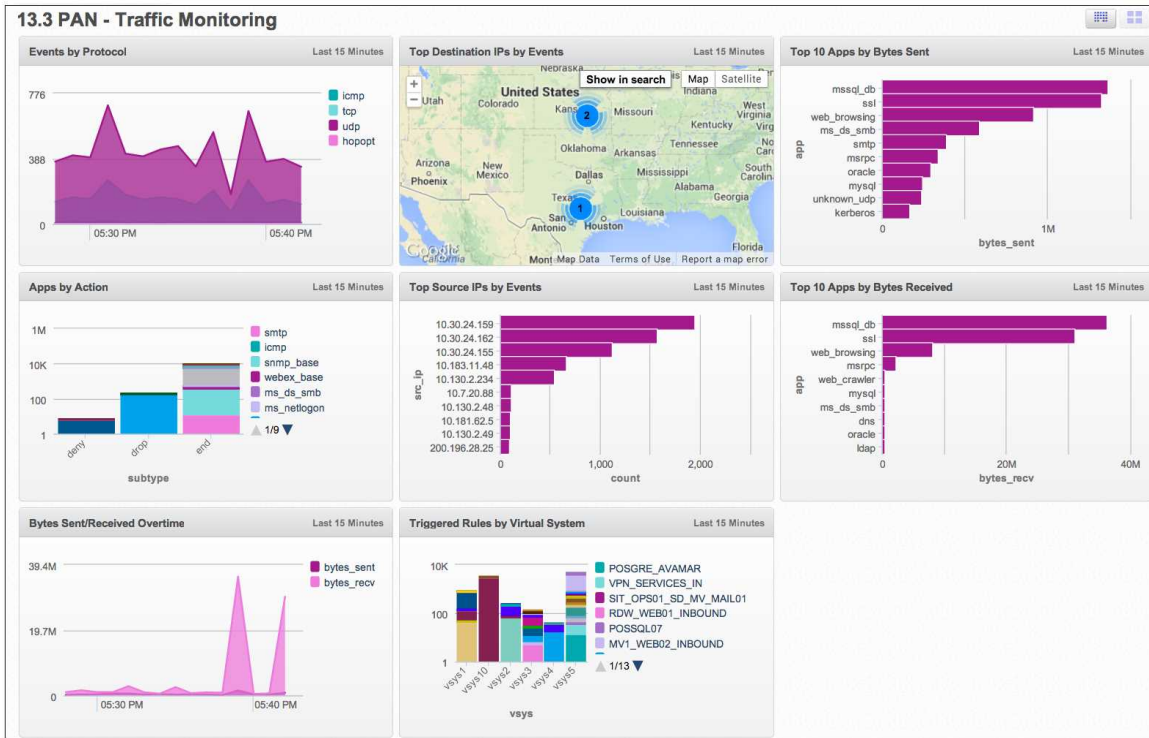
Threat by Category. The query behind this Monitor parses the threat ID and category from your Palo Alto Network logs, then returns the number of threats sorted by category.

Top Source IPs. Ranks the top 10 source IPs hitting your firewall.

App by Severity. Shows the breakdown of threats per app, sorted by threat level (Critical, High, Informational, and Low).

Traffic Monitoring Dashboard

The Traffic Monitoring Dashboard includes several Monitors that display information about incoming and outgoing traffic, including bytes sent and received.



Events by Protocol. Displays the breakdown of events, sorted by protocol (ICMP, TCP, UDP, HOPOPT).

Top Destination IPs by Events. Using a geolocation query, this Monitor maps which IPs are being accessed outside the network for all event types.

Top 10 Apps by Bytes Sent. Shows which apps are being sent the most bytes.

Apps by Action. This Monitor queries all traffic types and then displays each app per drop, denial, and success.

Top Source IPs by Events. Displays the top 10 IPs generating events.

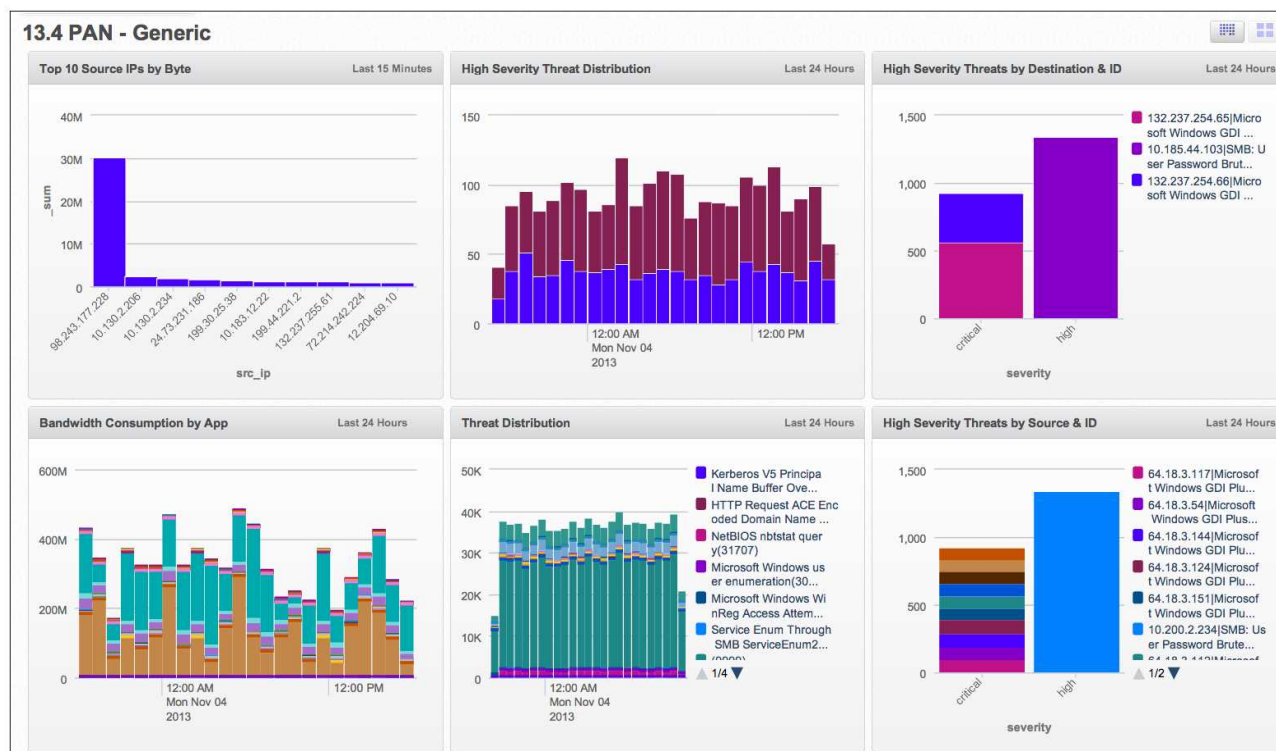
Top 10 Apps by Bytes Received. Traffic from the 10 most active apps is shown, making unexpected upticks in traffic easy to identify.

Bytes Sent/Received Overtime. Keep an eye on the overall inbound and outbound traffic in your deployment.

Triggered Rules by Virtual System. Including all existing trigger rules, this Monitor displays traffic from each virtual system in your deployment.

Generic Dashboard

This advanced Dashboard includes specialized, targeted Monitors that are typically used by IT Admins.



Top 10 Source IPs by Byte. Watch for unexpected spikes in traffic from the top 10 Source IP addresses.

High Severity Threat Distribution. Displays the severity of threats over the past hour.

High Severity Threats by Destination & ID. Counted by the number of threats coming from specific destinations and IP addresses, Critical and High severity threats are shown.

High Severity Threats by Source & ID. No need to guess where Critical and High threats are coming from. This Monitor displays each threat source.

Threat Distribution. Displays the source of threats as well as the number of threats over the past 24 hours.

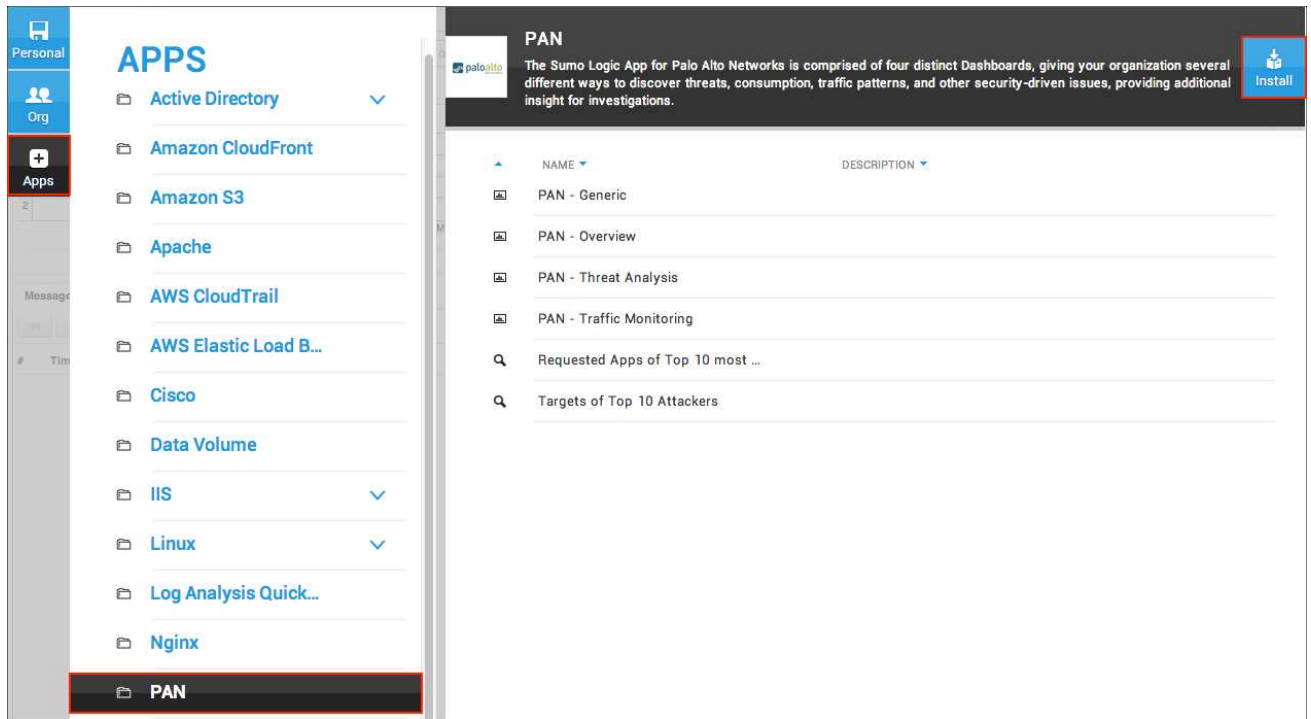
Bandwidth Consumption by App. View the total bandwidth consumed by each app in one place.

Installing the PAN App

The **Library** makes it easy to install the Sumo Logic Applications your organization needs.

To install the app:

1. In the **Library**, click the **Apps** tab.
2. Click **PAN**.
3. Click **Install**.



4. In the Install Application dialog box, do one of the following for **Data Source Mapping**:

- **Select from `_sourceCategory` values.** Choose an existing `_sourceCategory` present in your account used for PAN, such as **PaloAltoNetworks**, for example.

Important: If you do not select the correct `_sourceCategory`, data will not be loaded into the app. If you don't know which `_sourceCategory` to select, ask the administrator who configured the Source.

- **Custom data filter.** To set up a specific data filter, type the keyword(s) you'd like to use to filter the data.

PAN - Install Application

The Sumo Logic App for Palo Alto Networks is comprised of four distinct Dashboards, giving your organization several different ways to discover threats, consumption, traffic patterns, and other security-driven issues, providing additional insight for investigations.

Data Source Mapping

Data Source Mapping determines what logs are included in the application Dashboards and searches. You may select a source category or define a custom keyword search that maps the application to your organization's data set.

[Custom data filter help](#)

☒ Select from `_sourceCategory` values

PaloAltoNetworks

Resulting data filter

`_sourceCategory=PaloAltoNetworks`

☐ Custom data filter

Installation Location

Application Name*

PAN

Folder*

PERSONAL

Apache

Cancel | Install

5. For **Folder**, choose either the **Personal** folder or a subfolder in the **Personal** folder. (Click the blue + to create a new subfolder).
6. Click **Install**.

Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

Sumo Logic App for Varnish

Varnish is a web application accelerator placed between your users and the web servers. It acts as a caching solution for your web application and is designed to speed up delivery by a factor of 30-100x.

With the Sumo Logic Application for Varnish, you can:

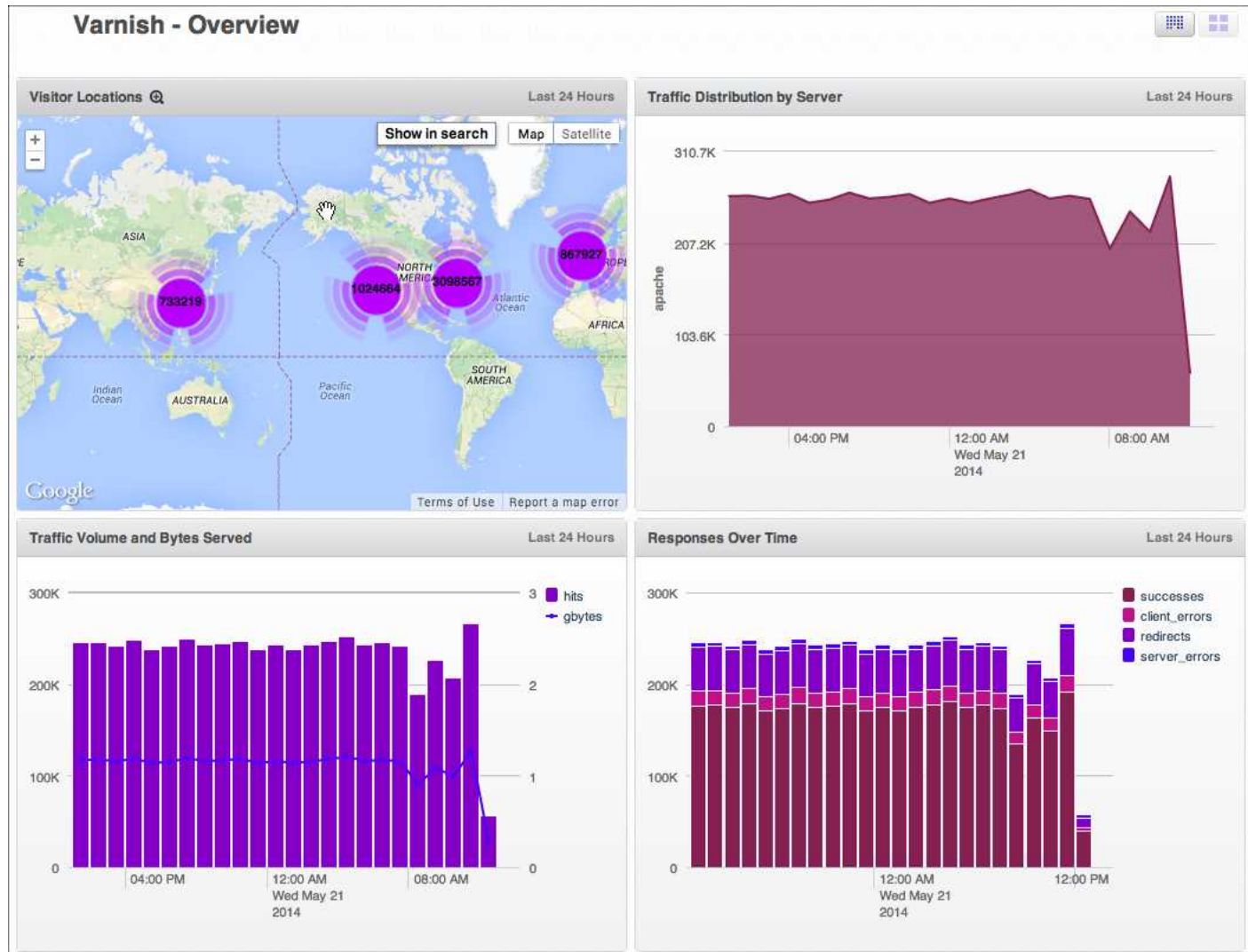
- Identify traffic sources and most requested products and quickly address both customer-facing and internal operation issues.
- Monitor and improve application and website work flow, and optimize your customer's paths and interactions.
- Understand how customers are using your product and service to help define future requirements.

The app uses predefined searches and Dashboards that provide visibility into your environment for real time analysis of overall usage.

Sumo Logic App for Varnish Dashboards

The Sumo Logic app for Varnish includes several Dashboards that allow you instant access to information about your system's visitors, traffic, and web server operations.

Varnish - Overview



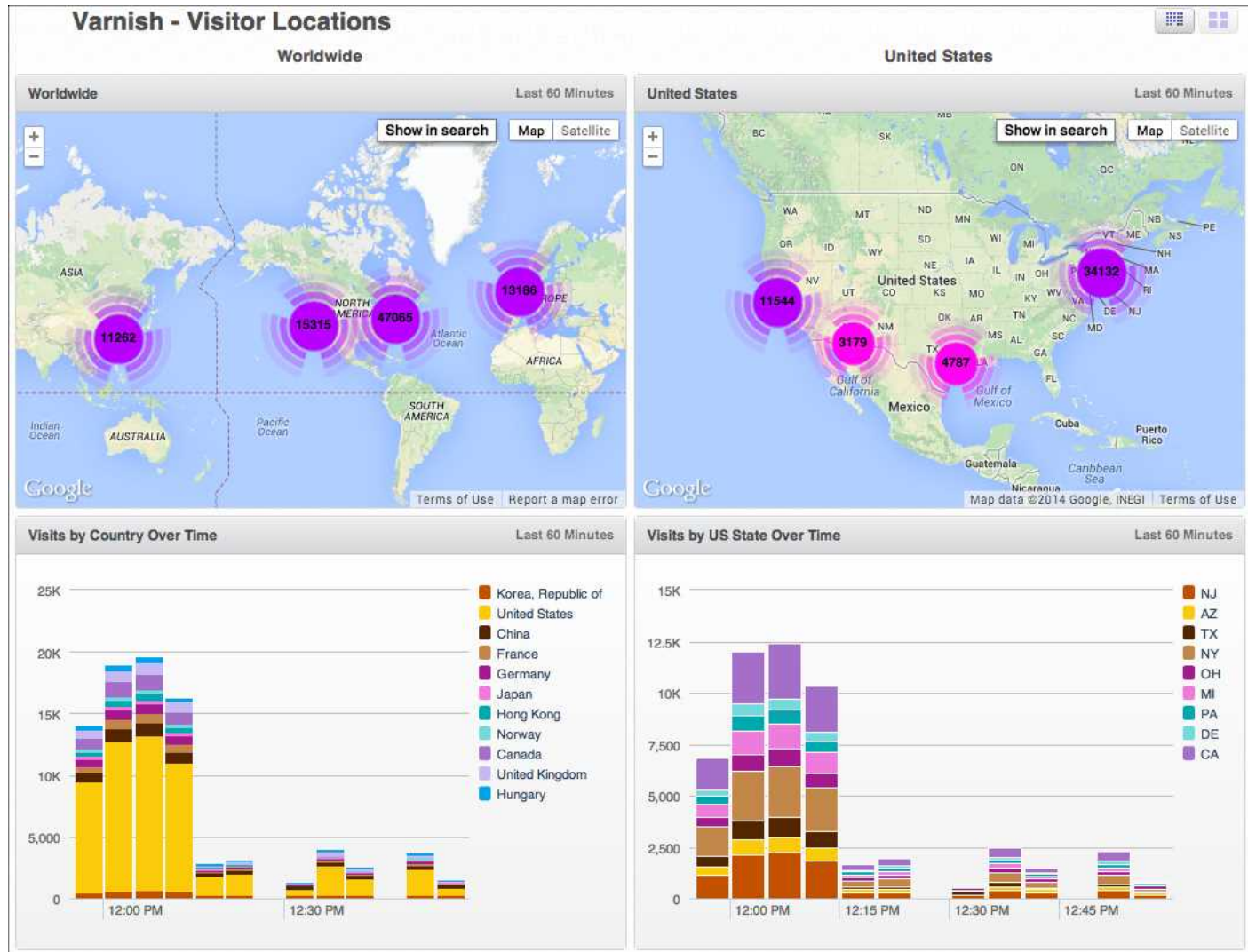
Visitor Locations. Performs a geo lookup operation and displays the number of visitors and their locations on a map of the world by IP address over the last 24 hours.

Traffic Volume and Bytes Served. Displays number of requests and web traffic volume in Gbytes using a combination column and line chart served in timeslices of one hour over the last 24 hours.

Traffic Distribution by Server. Shows how web traffic is distributed by server using an area chart in timeslices of one hour over the last 24 hours.

Responses Over Time. Provides information on the number of responses over time for successes, redirects, client errors, and server errors using a stacked column chart in timeslices of one hour for the last 24 hours.

Varnish - Visitor Locations



The Varnish Visitor Locations Dashboard provides information on Worldwide and United States visitors.

Worldwide. Performs a geo lookup operation and displays the number of worldwide visitors and their locations by IP address on a map of the world for the last hour.

Visits by Country Over Time. Displays the number of worldwide visitors to your site by country using a stacked column chart in timeslices of 5 minutes over the last hour.

United States. Performs a geo lookup operation and displays the number of U.S. visitors and their locations by IP address on a map of the world for the last hour.

Visits by US State Over Time. Displays the number of U.S. visitors to your site by state using a stacked column chart in timeslices of 5 minutes over the last hour.

Varnish - Visitor Access Types



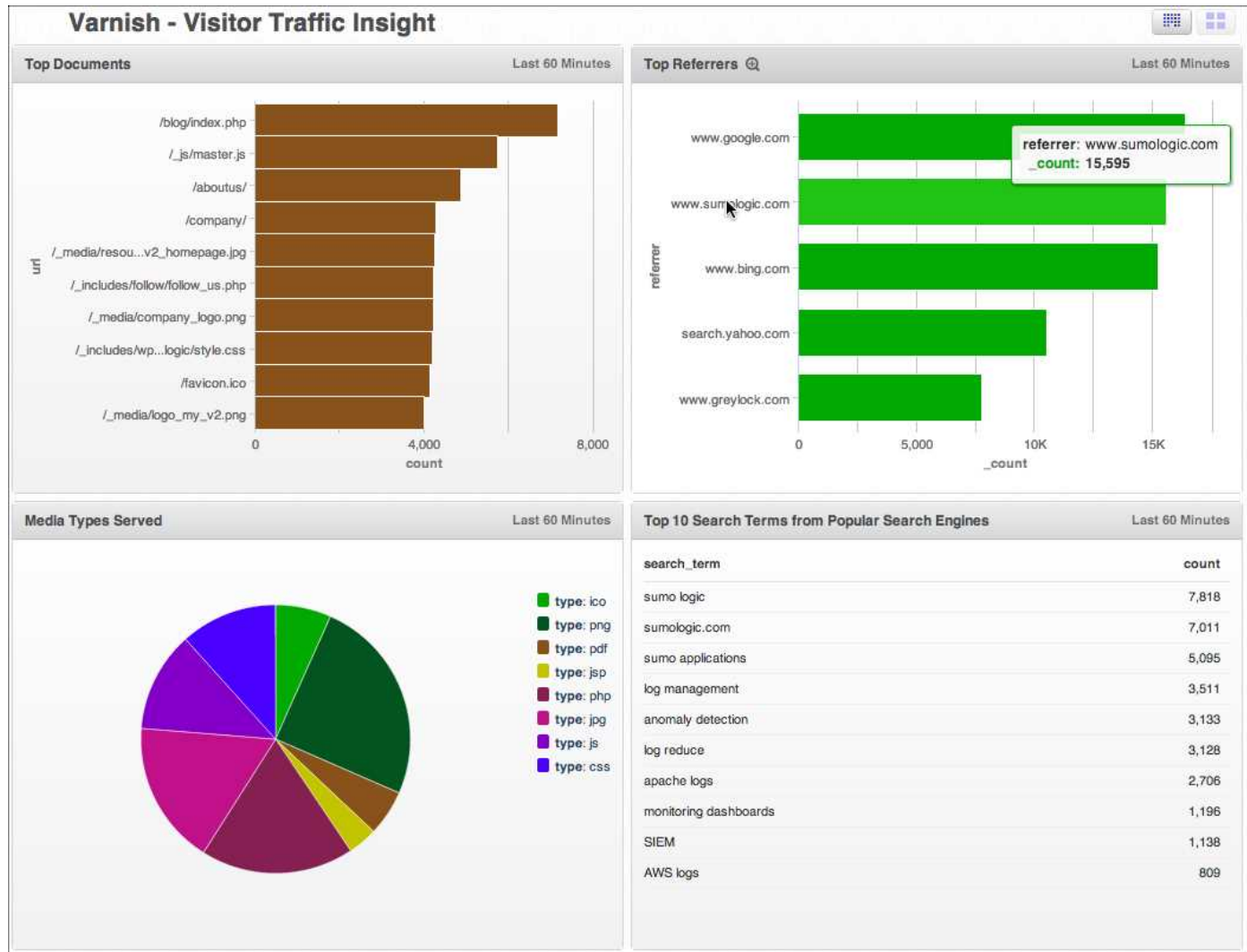
Visitor Platforms. Displays the platforms used by visitors to your site by type and percentage in a pie chart for the last hour.

Browsers and Operating Systems. Counts the browsers and operating systems used by your visitors and displays that information in a stacked column chart for the last hour.

Popular Mobile Device Versions. Displays the top 10 mobile device software versions in a column chart for the last hour. Hover over the column to display more information, including the software version number.

Top 10 PC and Mac Versions. Displays the top 10 PC and Mac software versions in a column chart for the last hour. Hover over the column to display more information, including the software version number.

Varnish - Visitor Traffic Insight



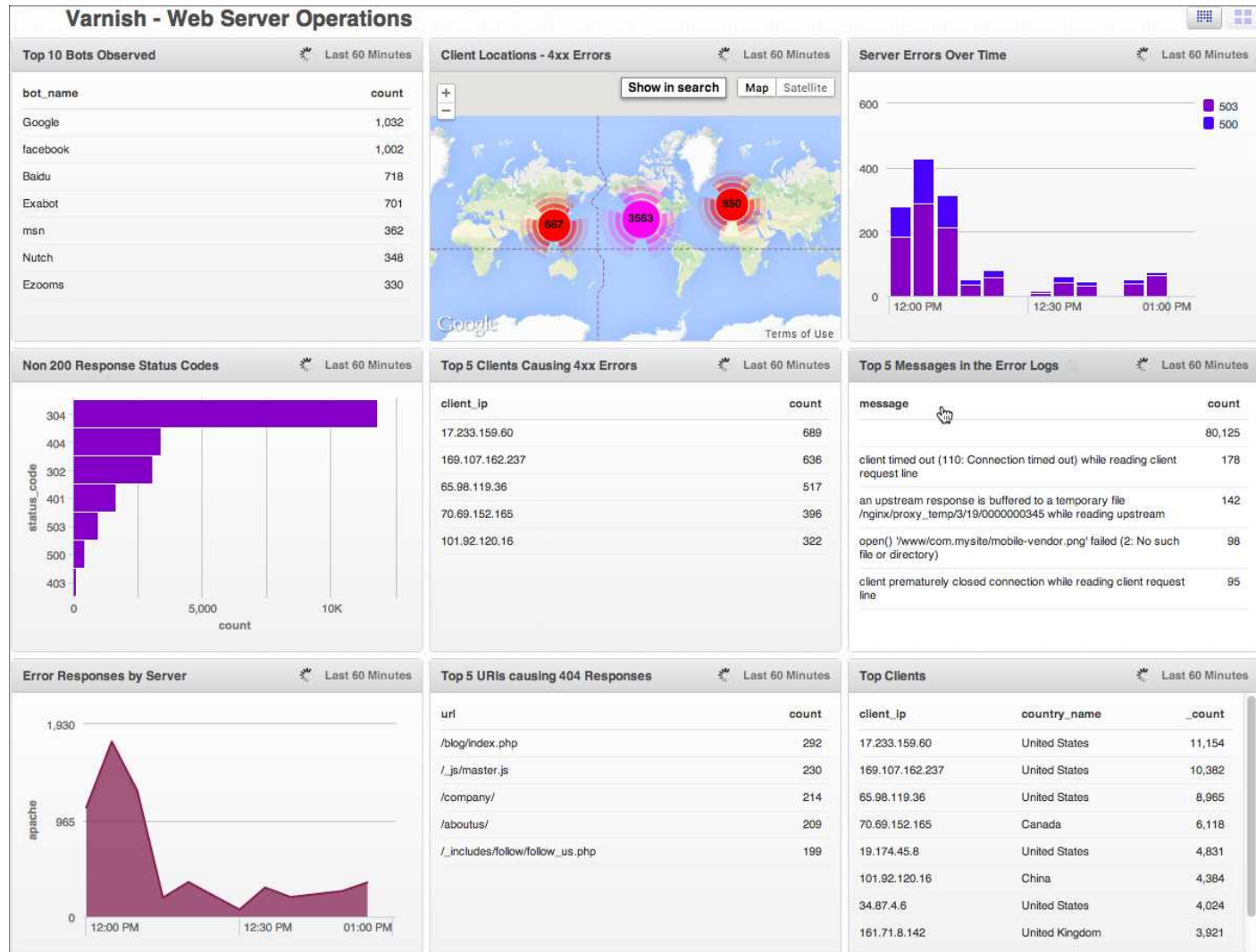
Top Documents. Displays the top 10 document URLs accessed by the most visitors to your site in a bar chart for the last hour. Hover over a section of the bar chart for details.

Media Types Served. Provides the percentage of file types accessed by visitors by URL in a pie chart for the last hour.

Top Referrers. Shows the top 5 referrers to your site in a bar chart for the last hour. Hover over a section of the bar chart for details.

Top 10 Search Terms from Popular Search Engines. Displays the top 10 search terms used by visitors on popular search engines by count for the last hour.

Varnish - Web Server Operations



Top 10 Bots Observed. Counts the top 10 web spider bots that have accessed your site in the last hour.

Non 200 Response Status Codes. Displays the non 200 error status codes that have been reported over the last hour in a bar chart.

Error Response by Server. Shows the error responses by your server for the last hour in timeslices of five minutes using an area chart.

Client Locations - 4xx Errors. A geo lookup query displays the IP addresses from where 4xx errors are originating on a map of the world for the last hour.

Top 5 Clients Causing 4xx Errors. Displays the top 5 client IP addresses where 4xx errors have originated in the last hour.

Top 5 URIs Causing 404 Responses. Lists the top 5 URIs that have caused 404 responses for the last hour.

Server Errors Over Time. Displays number of server errors and their status codes in timeslices of five minutes for the last hour using a stacked column chart.

Top 5 Messages in the Error Logs. Counts the top 5 most common messages in your system's error logs for the last hour.

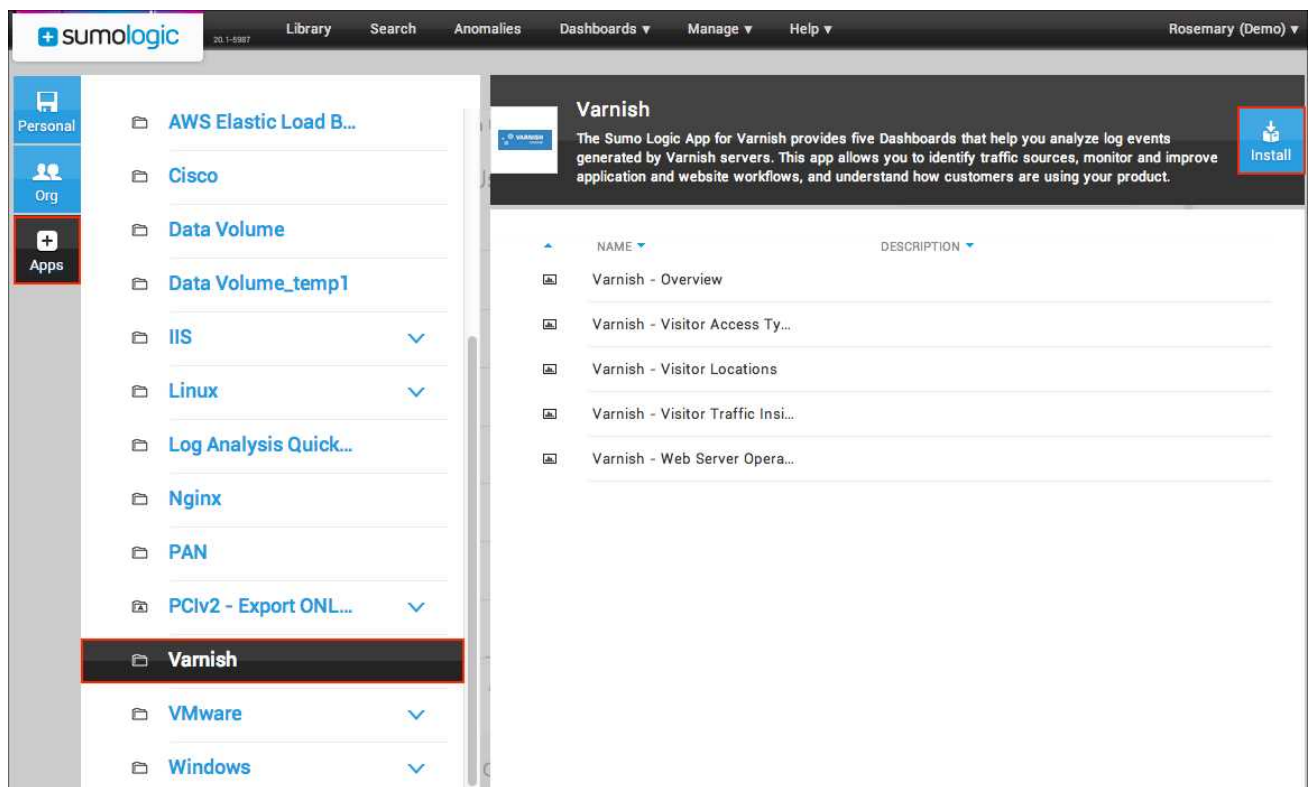
Top Clients. Displays the top 10 clients by IP address, country, and count for the last hour.

Installing the Varnish App

The **Library** feature of the Sumo Logic Web Application allows an Admin to install the Sumo Logic app for Apache. Your organization will be up and running with the app in just a few minutes.

To install the app:

1. In the **Library**, click the **Apps** tab.
2. Click **Varnish**.
3. Click **Install**.



4. In the **Install Application** dialog box, do one of the following for **Data Source Mapping**:
 - **Select from _sourceCategory values.** Choose an existing _sourceCategory present in your account used for Varnish.

Important: If you do not select the correct _sourceCategory, data will not be loaded into the app. If you don't know which _sourceCategory to select, ask the administrator who configured the Source.

- **Custom data filter.** To set up a specific data filter, type the keyword(s) you'd like to use to filter the data.

Varnish - Install Application

The Sumo Logic App for Varnish provides five Dashboards that help you analyze log events generated by Varnish servers. This app allows you to identify traffic sources, monitor and improve application and website workflows, and understand how customers are using your product.

Data Source Mapping

Data Source Mapping determines what logs are included in the application Dashboards and searches. You may select a source category or define a custom keyword search that maps the application to your organization's data set.

[Custom data filter help](#)

☒ Select from `_sourceCategory` values

Select a Category...

Resulting data filter

☐ Custom data filter

Installation Location

Application Name*

Varnish

Folder*

PERSONAL

Apache

Cancel | Install

5. For **Folder**, choose either the **Personal** folder or a subfolder in the **Personal** folder. (Click the blue + to create a new subfolder).
6. Click **Install**.

Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

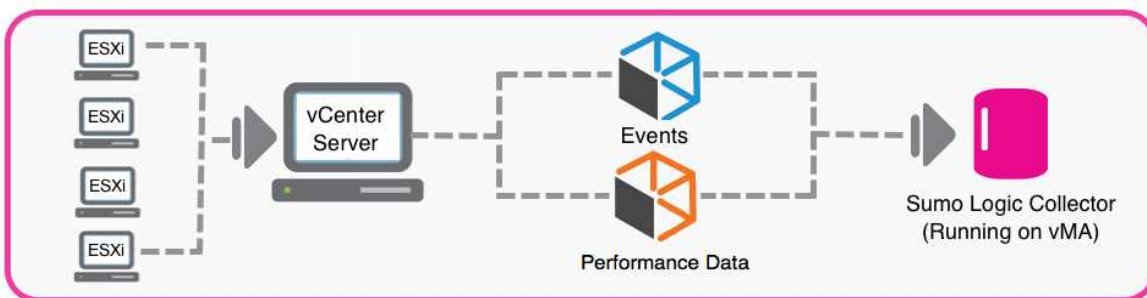
Sumo Logic App for VMware

The Sumo Logic Application for VMware allows you to:

- Collect and centralize logs from the entire VMware infrastructure, including ESX/ESXi, vCenter Server, and individual virtual machines as well as operating system and applications running within the virtual machine.
- Troubleshoot VMware farms and find issues related to over-provisioning VMs, "noisy neighbors", changes to configuration, and VM movement.
- Monitor the entire VMware infrastructure through real time dashboards that provide insight into key metrics such as VM CPU, memory and disk utilization. Determine capacity constrained and under-provisioned physical hosts and idle VMs to improve deployment strategy and optimize cost.

Collecting logs for the Sumo Logic Application for VMware

The logs collected from vCenter Servers enables you to use the Sumo Logic Application for VMware search, visualize, and analyze vCenter Server Events and Performance Data in real time to enable monitoring and detect important events within your virtual environment.



Setting up a vMA Server to Collect Data

Before setting up a source to collect data, you'll need to install vMA through the vCenter Server (if it's not already installed) and then download and install a Collector.

Step 1: Install vMA

vMA is an appliance (SUSE virtual machine) that includes vSphere CLI, and vSphere SDK for Perl. It allows administrators to run scripts or agents that interact with ESXi hosts and vCenter Server systems without having to authenticate each time.

To set up vMA:

1. Download vMA from VMware, and follow the accompanying instructions to deploy the OVF. (Go to the vCenter Server through vSphere client, then choose File > Deploy OVF Template).



Because vSphere 5.1 has known issues with resxtop and SSL certifications, be sure to use vSphere 5.0, 5.0.0.1 or 5.0.0.2.

(The known issues in 5.1 are documented in the [Release Notes](#).)

2. Setup the authentication, timezone and time for the vMA by following instructions in the VMware documentation.
3. Run `credstore_admin.pl` on the vMA to add credentials for each vCenter server that generates performance and event data you'd like to collect. This script comes by default with vSphere Perl SDK; on vMA it's located under `/usr/lib/vmware-vcli/apps/general`. (See [this VMware KB article](#) for more information). For example, to add a user's account on a vCenter Server you can run:

```
/usr/lib/vmware-vcli/apps/general/credstore_admin.pl
```

Run the following command to verify that authentication is set up correctly and to see a list of network interfaces:

```
esxcli --server <vcenter host> --vihost <esxi host> network nic list
```

For example, if we run `esxcli --server 192.168.23.242 --vihost 192.168.23.24 network nic`, we'll see a list of network interfaces for the ESXi host (192.168.23.24) managed by our vCenter Server (192.168.23.242).

Step 2: Download and Install the Collector on vMA

1. Sign in to [the Sumo Logic Web Application](#). Click the **Collectors** tab. You will see a link to **Download Collector** at the top right of the tab. Click the link, and then copy the download URL for the installer file you need.
2. On the vMA machine, use `wget` or `curl` to download the file from the URL. The URL must be enclosed in double-quotes to work with `wget`.
3. From the download directory, run the installation file `SumoCollector_[os-type]_[build-date]_xxxxx.sh` as root. First, make sure root has executable privileges for the file by typing:

```
chmod 740 SumoCollector_[os-type]_[build-date]_xxxxx.sh
```

4. Run the install file on your server with root privileges:

```
$ sudo ./SumoCollector_[os-type]_[build-date]_xxxxx.sh
```

The Collector runs as a service and starts automatically after installing or rebooting.

Collecting Event Messages

An event is an action that triggers an event message on a vCenter Server. Event messages are not logged, but are instead stored in the vCenter Server database. Sumo Logic for VMware retrieves these messages using the vSphere Perl SDK that comes with vMA (by default).

Step 1: Configure a Syslog Source for the Collector.

A Sumo Logic Syslog Source operates like a syslog server listening on the designated port to receive Syslog messages.

1. In the Web Application, click the **Collectors** tab, and then click **Add Source**.
2. Select **Syslog** for the Source type.



3. Enter a **Name** to display for this Source. Source name metadata is stored in a searchable field called `_sourceName`.
4. For **Protocol** choose **TCP**.
5. Enter the **Port** number for the Source to listen to (for example, 1514, but choose the correct port for your Collector).
6. For **Source Category**, we recommend typing `vcenter_log`.
7. Under **Advanced**, set the following options:
 - Select **Extract timestamp information from log file entries**.
 - Select **Ignore time zone from log file and instead use** and then choose **UTC** from the menu (as shown below).

The screenshot shows the configuration form for a Syslog Source. The "Name" field is filled with "vCenter Server Collector". The "Protocol" is set to "TCP" and the "Port" is "1514". The "Source Category" is "vcenter_log". The "Advanced" section is expanded, showing the "Enable Timestamp Parsing" checkbox checked. Under "Time Zone", the "Ignore time zone from log file and instead use:" option is selected, and the dropdown menu shows "(UTC) UTC". The "Save" and "Cancel" buttons are at the bottom right.

9. Click **Save**.

Step 2: Configure Logs to be Collected

1. On the vMA, create a directory to hold all Sumo Logic scripts attached found [here](#). You might want to name the directory `/var/log/vmware` or something similar.
2. Download the scripts, then put them in the directory you just created.
3. Edit the `cron_vcenter_events.sh` script by changing the `SCRIPT_PATH` variable to reflect the absolute path where the script resides.



If you have multiple vCenter servers, create a new line for each one. Make sure you add the credential for each server (as described in [Step 1: Install vMA](#)).

4. Test the running of the `query_vCenter.pl` script (that queries the vCenter Server for events) as described in [Troubleshooting and Manual Testing](#).
5. Create a cron job to periodically run the `cron_vcenter_events.sh` script at the interval you'd like. (**Note:** You'll need to have the `LD_LIBRARY_PATH` env variable in the crontab line.) For example, to run the job every two minutes, you'd use something like:

```
*/2 * * * * LD_LIBRARY_PATH=/opt/vmware/vma/lib64:/opt/vmware/vma/lib
/var/log/vmware/SumoLogic/cron_vcenter_events.sh
```

Collecting Performance Logs

Collecting performance logs involves using VMware tools and scripts running on vMA to extract performance statistics.

Step 1: Configure a Local File Source.

A Sumo Logic Local File Source collects from a local file.

1. In the Web Application, click the **Collectors** tab, and then click **Add Source** for your vCenter Server Collector.
2. Select **Local File** for the Source type.
3. Enter a **Name** to display for this Source. Source name metadata is stored in a searchable field called `_sourceName`.
4. For File Path, enter `*.perf.out`.
4. For **Source Category**, type `esx_perf`.
5. Under **Advanced**, make sure that **Timestamp Parsing** is selected. Then for **Time Zone** choose the time zone of the vMA virtual machine.

Manage Collectors and Sources

Collectors and Sources > Add a Source

Select a type of Source:

Local File

Collects any local files except Windows Event logs and Windows Performance Monitor logs.

Remote File

Collects any remote files except Windows Event logs and Windows Performance Monitor logs.

Syslog

Collects syslog messages streaming to a syslog port.

Windows Event Log

Collects Windows Event logs from a Windows system.

Name* vCenter
Maximum name length is 128 characters

Description

File Path* *.perf.out
Absolute path expression to one or more files, or Windows UNC Share path for Windows collectors only.
For example: /varlog/messages or /varlog/*.log or \\hostname\path\to\directory

Collection should begin 24 hours ago
(starts approx. at 4/6/2014 12AM)

Source Host
Host name for the local machine, e.g. LDAP_Server

Source Category esx_perf
Log category metadata to use later for querying, e.g. OS_Security

Advanced

Blacklist
One or more comma separated path expressions describing the files to be excluded.
For example: /varlog/***.bak, /varfoldlog/*.log

Enable Timestamp Parsing ☒ Extract timestamp information from log file entries

Time Zone ☐ Use time zone from log file. If none is present use:
(UTC) UTC

☒ Ignore time zone from log file and instead use:
(GMT-08:00) Pacific Time (US & Canada)

Timestamp Format ☒ Automatically detect the format ☐ Specify a format

Enable Multiline Processing ☐ Detect messages spanning multiple lines

☒ Infer Boundaries - Detect message boundaries automatically
Please note, Infer Boundaries may not be accurate for all log types.

☐ Boundary Regex - Expression to match message boundary e.g. (?<R)(\r+)

Filters

Save | **Cancel**

6. Click **Save**.

Step 2: Configure Performance Logs for Collection

Before collecting can begin, you'll need to invoke scripts to transform the performance data from the **resxtop** utility so it's delivered in a format that Sumo Logic can consume.

1. On the vMA, create a directory to hold all scripts (say /var/log/vmware).
2. Extract all files from the Zip bundle provided by Sumo Logic to the directory you just created.

3. Edit the **vcenter.info** file so one vCenter Server and one username is on each line. For example:

```
vcenter01.company.com root
vcenter02.company.com root
vcenter03.company.com root
```

You can run **/usr/lib/vmware-vcli/apps/general/credstore_admin.pl list** to get a list of all the vCenter Servers you have already configured for authentication.

4. Edit the following in the **cron_vcenter_perf.sh** script:
 - Change the **SCRIPT_PATH** variable to reflect the absolute path where the script resides.
 - Select the method you'd like to use to collect performance data. Then, uncomment the line that calls **\$\$SCRIPT_PATH/getserver_perf.pl**. For more information, see [Segmenting Collection](#).
6. Run the **cron_vcenter_perf.sh** script. After it finishes, verify that performance logs are being collected. You should see the **esxi.perf.out** file in the above **\$\$SCRIPT_PATH** directory; the file should have at least 7000-8000 messages per ESXi server that is managed by a vCenter.



The script collects all the default performance data returned by the **resxtp** utility. If however, you need to collect only certain performance related data, then invoke **resxtp** on the VMA machine in interactive mode, select only the panels and columns that you want to collect, and then save the configuration in the **/home/vi-admin/.esxtp50rc** file.

7. Create a cron job to periodically run the **cron_vcenter_perf.sh** script. For example to run the script every 15 minutes, it would look like:

```
*/15 * * * * LD_LIBRARY_PATH=/opt/vmware/vma/lib64:/opt/vmware/vma/lib /var/log/vmware/cron_
vcenter_perf.sh
```

Understanding vCenter Scripts

In this section we'll walk you through a few important tasks that involve the **query_vCenter.pl** script, which queries the vCenter Server for events.

Troubleshooting and Manual Testing

It's a good idea to test the **query_vCenter.pl** script before setting up a CRON job.

To test the **query_vCenter.pl** script:

- Go to the folder that holds all the scripts (for example, **/var/log/vmware**) and run:

```
query_vcenter.pl -s [vcenterserver] -f output.txt
```

(Replace **[vCenterServer]** with the name of the target vCenter Server in your environment.)

In the standard output, you should see the query time range, the number of events collected. The events themselves are stored inside the **output.txt** file. If you're prompted to enter a username or password it means that the credentials for the target vCenter Server are not set properly. By default, the first time **query_vCenter** is called,

events from the past 24 hours are collected. If you want to start collecting events older than the past 24 hours, please see the section [below](#).



Because the above information is logged into `/var/log/message` for cron jobs, it's a good idea to monitor `/var/log/messages` to make sure the collection CRON jobs work well.

Collecting Historical Events

By default, the first time `query_vCenter.pl` is called, events from the past 24 hours are collected. Each time the script is called, it writes the timestamp of the last read event in a file named `.timelog` for the next call to pick up.

If you want to start collecting events older than the past 24 hours, before setting up the CRON job for `cron_vcenter_events.sh`, do the following on the VMA machine.

To collect historical events:

1. Go to the vi-admin home directory, at `/home/vi-admin`.
2. Set the `SCRIPT_PATH` environment variable to point to where all the Sumo Logic for VMware scripts reside. For example:

```
vi-admin@vma1:~> export SCRIPT_PATH=/var/log/vmware
```

3. Run the `query_vCenter.pl` script as follows:

```
./query_vCenter.pl --server <vcenter server> --target <syslog host>:<syslog port> --bT <time in UTC>
```



The syslog server should reflect the IP address or hostname of the vMA machine; the `syslog_port` should reflect the port number that you previously set up for the Source responsible for collecting vCenter Server Events.

For example, to collect all events starting from 5:00pm on October 8, 2012 in the UTC timezone, you would run the following command:

```
./query_vCenter.pl --server 192.168.23.242 --target vmahost:1514 --bT 2012-10-08T00:17:00.00Z
```

Once this command completes successfully, you can begin to pick up ongoing events by setting up the CRON job as described in step 2 of [Collecting Event Messages](#).

Segmenting Performance Collection

Performance data collection for ESXi servers associated with a vCenter server works by sequentially getting data from each ESXi server. Having a large number of servers associated with a vCenter can cause delays. To avoid these delays you can **parallelize** collection by creating multiple **segments** from the list of ESXi servers associated with a vCenter server. The number of segments would depend on the amount of data you are collecting and how often you would like to collect performance data.

For example, let's say you have a 100 ESXi servers associated with a single vCenter server; it takes more than 2 minutes for the performance collection script to collect data from all ESXi servers. If you need to collect performance data snapshots every 2 minutes, you'd segment the collection into two or more instances of the `cron_vcenter_perf.sh` script to parallelize collection.

Or, if you have 100 ESXi servers and five segments, then each segment would have 20 servers. However, if there's an odd number of servers, say 145 servers and six segments, then five segments have 24 servers, and the last (sixth) segment will have 20 servers.

The total number of segments is specified by the `-segments` option and the segment number we want to run is specified by the `segment_number` option.

Before collecting from multiple segments, you can test how ESXi servers will be divided up into segments by using the `-test` option. For example, to test which servers get assigned to each segment assuming you have two segments, you would run the following command:

```
$SCRIPT_PATH/getserver_perf.pl -test -type=vcenter -path=$SCRIPT_PATH -server_file=$SCRIPT_PATH/vcenter.info -output_file=$SCRIPT_PATH/vcenter_segments.perf.out -segments=2 -segment_number=1 | /bin/logger
```

Then to get performance data just for segment one, you can run the following command:

```
$SCRIPT_PATH/getserver_perf.pl -type=vcenter -path=$SCRIPT_PATH -server_file=$SCRIPT_PATH/vcenter.info -output_file=$SCRIPT_PATH/vcenter_segment-3.perf.out -segments=2 -segment_number=1 | /bin/logger
```

and so on.

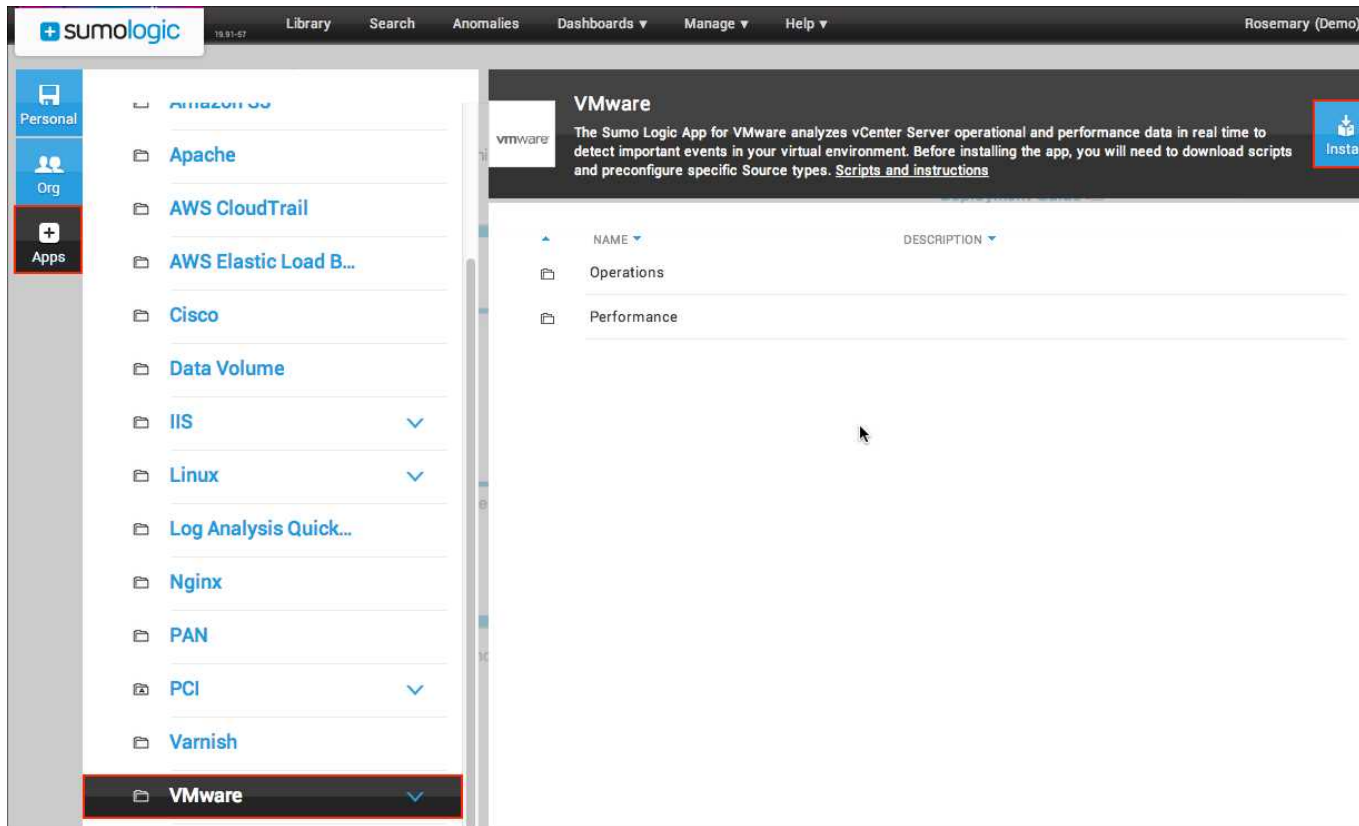
Once you are satisfied with the time it takes for collecting data for a segment, create multiple scripts to collect data for each segment based on the `cron_vcenter_perf.sh` script. Schedule these scripts to run as a cron job according to the desired frequency.

Installing the app for VMware

After [configuring the collection](#) of VMware logs, you can install the Sumo Logic App for VMware to begin using the Dashboards and searches in the app.

To install the app:

1. In the **Library**, click the **Apps** tab.
2. Click **VMware**.
3. Click **Install**.



4. In the **Install Application** dialog box, select **esx_perf** from the **_sourceCategory** menu. If you don't see this option, please make sure you've configured VMware log collection properly. **Important:** If you do not select the correct **_sourceCategory**, data will not be loaded into the app. If you don't know which **_sourceCategory** to select, ask the administrator who configured the Source.
5. For **Custom data filter**, type (**_sourceCategory=esx_perf OR _sourceCategory=vcenter_log**).

6. Click Install.

VMware - Install Application

The Sumo Logic App for VMware analyzes vCenter Server operational and performance data in real time to detect important events in your virtual environment. Before installing the app, you will need to download scripts and preconfigure specific Source types. [Scripts and instructions](#)

Data Source Mapping

Data Source Mapping determines what logs are included in the application Dashboards and searches. You may select a source category or define a custom keyword search that maps the application to your organization's data set.

[Custom data filter help](#)

☐ Select from _sourceCategory values

esx_perf

Resulting data filter

_sourceCategory=esx_perf

☒ Custom data filter

(_sourceCategory=esx_perf OR _sourceCategory=vcenter_log)

Installation Location

Application Name*

VMware

Folder*

PERSONAL

Apache

Cancel | **Install**

Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

Sumo Logic App for Windows

The Sumo Logic application for Windows provides insight into your system's operation and events so that you can better manage and maintain your Windows systems. The app uses predefined parsers, searches, and Dashboards that provide visibility into your environment for real time analysis of overall usage.

The Sumo Logic app for Windows consists of two Dashboards and an extensive set of searches grouped by their topics under four sub-folders: Security Status, System Activity, Updates, and User Activity.

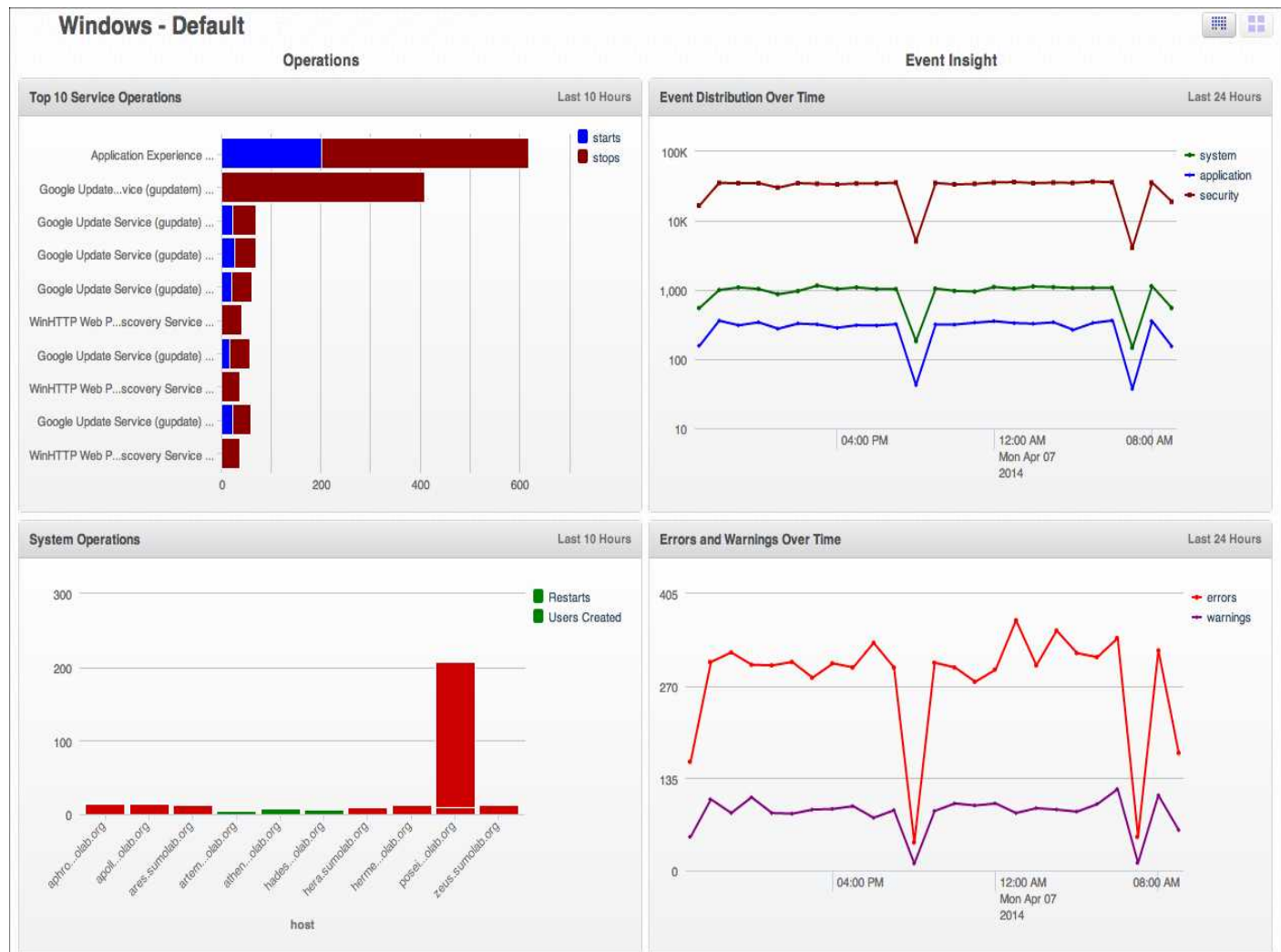
Log Types

The Sumo Logic application for Windows assumes events are coming from remote Windows Event Log Sources configured in Sumo Logic Collectors. It does not work with third party logs.

Sumo Logic App for Windows

The Sumo Logic app for Windows consists of two Dashboards and an extensive set of searches grouped by their topics under four sub-folders: Security Status, System Activity, Updates, and User Activity.

Default Dashboard



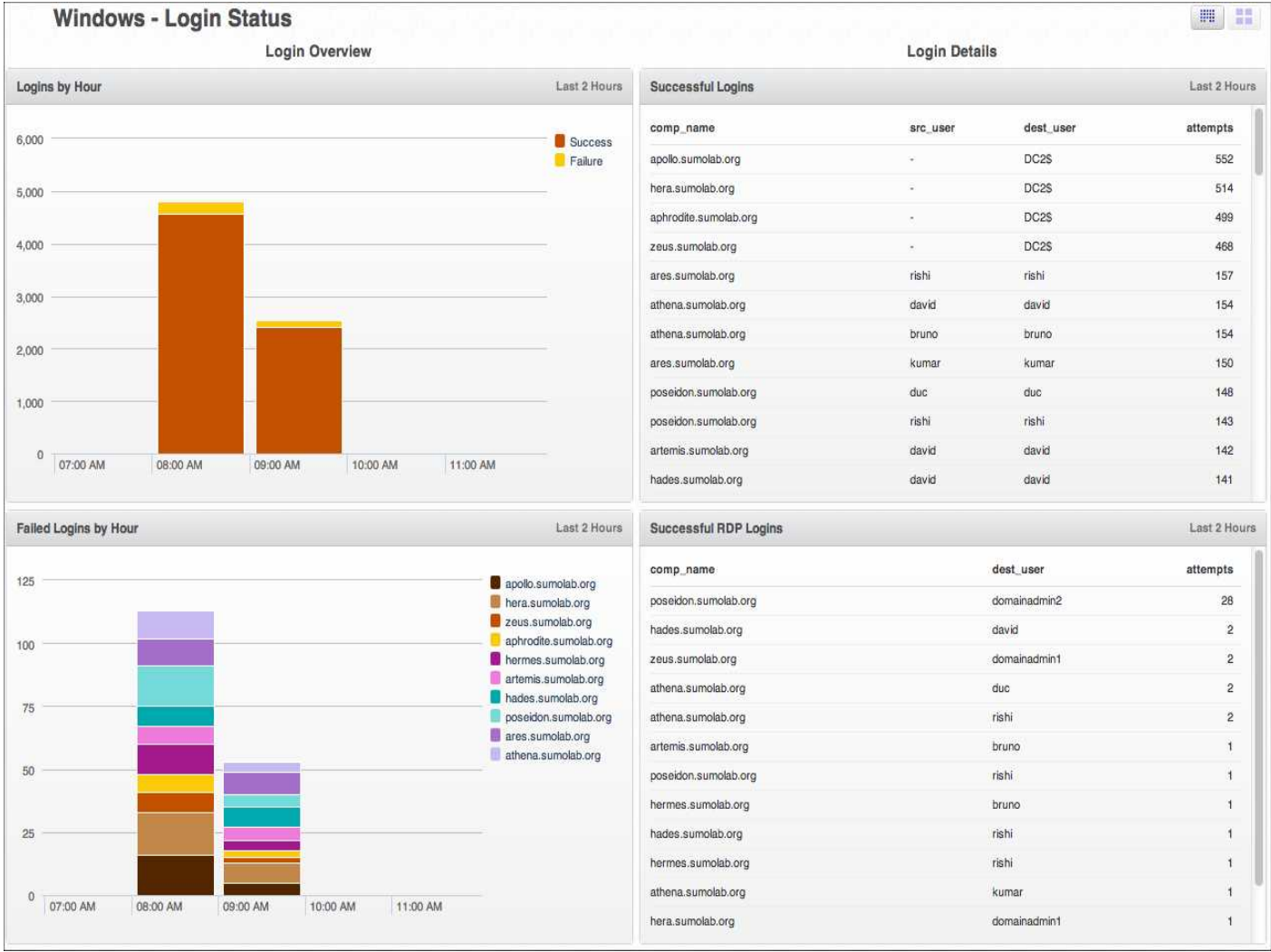
Top 10 Service Operations. Displays information on the top 10 services per host that have started and stopped over the last 10 hours in a bar chart. To display details of the data in a pop-up menu, hover over a section of the chart. Hover over the text **Last 10 Hours** in the upper right corner to see details of the time frame for the displayed data.

System Operations. Provides information on the number of and type of events that have occurred per host over the last 10 hours, which allows you to easily identify any spikes in activity in the column chart. To display details of the data in a pop-up menu, hover over a section of the chart. Hover over the text **Last 10 Hours** in the upper right corner to see details of the time frame for the displayed data.

Event Distribution Over Time. Displays the number and type of events per hour in an easy to read timeline for the past 24 hours. To display details of the data in a pop-up menu, hover over a line in the chart. Hover over the text **Last 24 Hours** in the upper right corner to see details of the time frame for the displayed data.

Errors and Warnings Over Time. Shows the number of errors and warnings per hour in a timeline. To display details of the data in a pop-up menu, hover over a line in the chart. Hover over the text **Last 24 Hours** in the upper right corner to see details of the time frame for the displayed data.

Login Status Dashboard



Logins by Hour. Counts the number of login successes and failures by one hour increments over the last two hours in a column chart. To display details of the data in a pop-up menu, hover over a section of the chart. Hover over the text in the upper right corner, **Last 2 Hours**, to see details of the time frame for the displayed data.

Failed Logins by Hour. Displays the number of failed logins per host by hour in an easy to read stacked column chart. Information is displayed for the last two hours. To display details of the data in a pop-up menu, hover over a section of the chart. Hover over the text in the upper right corner, **Last 2 Hours**, to see details of the time frame for the displayed data.

Successful Logins. Shows a table of successful logins including information on the computer name, source user, destination user, and number of attempts. Information is displayed for the last two hours.

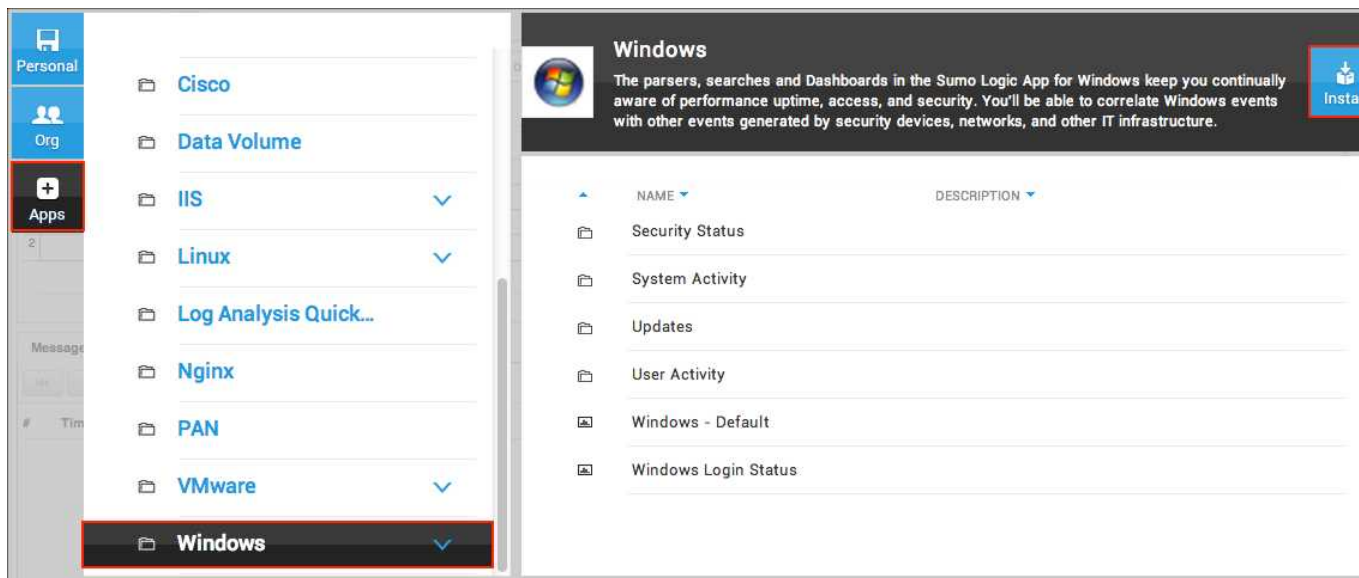
Successful RDP Logins. Provides a table with a list of successful remote desktop logins including details on computer name, destination user, and number of attempts. Information is displayed for the last two hours.

Installing the Sumo Logic App for Windows

The Library feature of the Sumo Logic Web Application allows an Admin to install the Windows App. Your organization will be up and running with the app in just a few minutes.

To install the app:

1. In the **Library**, click the **Apps** tab.
2. Click **Windows**.
3. Click **Install**.



4. **Select from `_sourceCategory` values.** Choose an existing `_sourceCategory` present in your account that is associated with logs from remote Windows Event Log sources.

Important: If you do not select the correct `_sourceCategory`, data will not be loaded into the app. If you

don't know which `_sourceCategory` to select, ask the administrator who configured the Source.

Windows - Install Application

The parsers, searches and Dashboards in the Sumo Logic App for Windows keep you continually aware of performance uptime, access, and security. You'll be able to correlate Windows events with other events generated by security devices, networks, and other IT infrastructure.

Data Source Mapping

Data Source Mapping determines what logs are included in the application Dashboards and searches. You may select a source category or define a custom keyword search that maps the application to your organization's data set.

[Custom data filter help](#)

☒ Select from `_sourceCategory` values

OS/Windows

Resulting data filter

`_sourceCategory=OS/Windows`

☐ Custom data filter

Installation Location

Application Name*

Windows

Folder*

PERSONAL

Apache

Cancel | Install

5. Click **Install**.

Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

Sumo Logic App for Windows Active Directory

The Sumo Logic Application for Windows Active Directory allows you to analyze Windows Active Directory logs and gain insight into your deployment. Using the app, you can identify user activity across your network and security administration systems.

The app uses predefined searches and Dashboards that provide visibility into your environment for real time analysis of overall usage.

Collecting Active Directory log files

Windows Active Directory (AD) is a directory service developed by Microsoft that stores information about various objects on a network. The Sumo Logic Application for Active Directory analyzes, then graphically displays this information to users and network administrators, including information about domain controllers, forest, site, users, groups, computers and organizational units. Sumo Logic users can augment or couple regular Windows Events with this data to get more contextual insights from the logs. For example, by augmenting the events based on the domain name, users can build searches specific to a specific AD site, or can track activities to users under a specific Organizational Unit.

Verifying Active Directory Module

Before proceeding, you'll need to verify that the Active Directory is available, which is a prerequisite for installation. The Active Directory Module is supported on Windows 7, and on Windows 2008 Server (R2 and later) if Remote Server Administration Tools (RSAT) is installed. You'll find more information from Microsoft [here](#).

To verify that Active Directory Module is available:

1. Choose **Start > Administrative Tools**.
2. Look for **Active Directory Module for Windows PowerShell**.



3. If the module isn't installed, install RSAT as described [here](#).

Deploying Sumo Logic scripts

In order to collect files, you'll need to download scripts found [here](#).

The package of Sumo Logic scripts contains the following:

- adObjectCollector.ps1
- adQueryDS.ps1
- domainCollector.ps1

networkUtils.ps1 contains core functions, which are leveraged by the other two scripts. After deploying the scripts, you'll need to configure a Script Source for domainCollector.ps1 and another Script Source for adObjectCollector.ps1. These scripts should be deployed on a machine that is part of the domain where the log files exist.

To deploy the scripts:

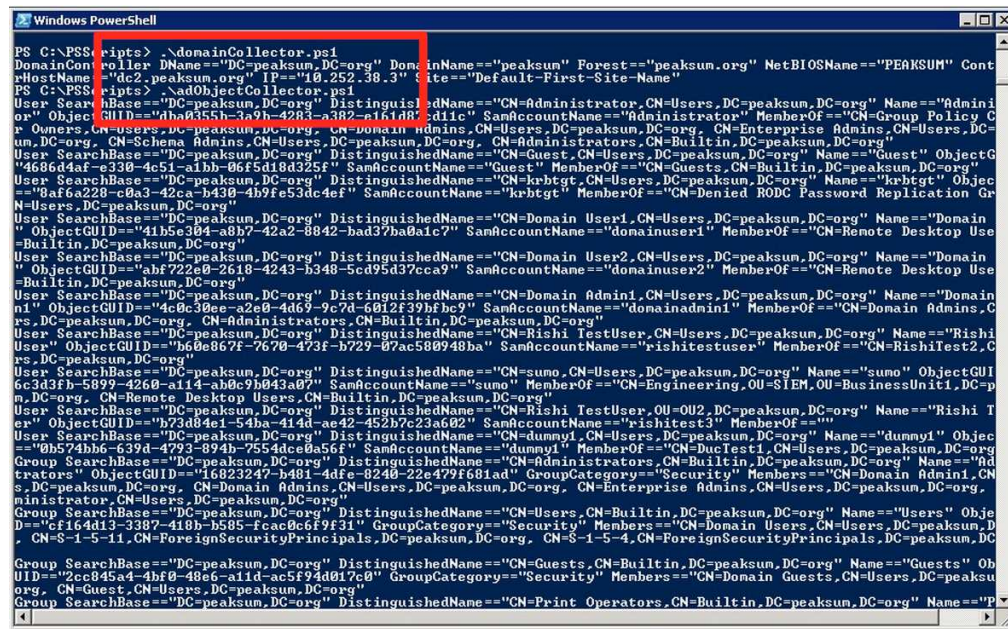
- Download the scripts to a folder, for example "C:\PSScripts"
Make sure the SCRIPTPATH inside these files match the path to the folder.

(Optional) To manually test the scripts:

1. Open a command line interface.
2. Run **domainCollector** and **adObjectCollector**:

```
powershell.exe -ExecutionPolicy Bypass -InputFormat None -File
c:\PSScripts\domainCollector.ps1
powershell.exe -ExecutionPolicy Bypass -InputFormat None -File
c:\PSScripts\adObjectCollector.ps1
```

If the setup has been successful, Active Directory domain and object information is collected, and the scripts print out results to the screen, similar to:



```
PS C:\PSScripts> .\domainCollector.ps1
DomainCollector: DNName="DC-peaksun,DC-org" DomainName="peaksun" Forest="peaksun.org" NetBIOSName="PEAKSUM" Cont
HostName="10.2.252.38" IP="10.252.38.3" Site="Default-First-Site-Name"
PS C:\PSScripts> .\adObjectCollector.ps1
User SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=Administrator,CN=Users,DC-peaksun,DC-org" Name="Admini
r Owners,CN=Users,DC-peaksun,DC-org, CN=Domain Admins,CN=Users,DC-peaksun,DC-org, CN=Enterprise Admins,CN=Users,DC-
un,DC-org, CN=Schema Admins,CN=Users,DC-peaksun,DC-org, CN=Administrators,CN=Builtin,DC-peaksun,DC-org"
User SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=Guest,CN=Users,DC-peaksun,DC-org" Name="Guest" ObjecG
"4686d4af-e330-4c51-a1bb-06f5d18d325f" SamAccountName="Guest" MemberOf="CN=Guests,CN=Builtin,DC-peaksun,DC-org"
"8af6a220-c0d3-42ca-b430-4b7fe53dc4ef" SamAccountName="krbtgt" MemberOf="CN=Denied RODC Password Replication Gr
N=Users,DC-peaksun,DC-org"
User SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=Domain User1,CN=Users,DC-peaksun,DC-org" Name="Domain
" ObjectGUID="41b5e304-a8b7-42a2-8842-bad37ba0a1c7" SamAccountName="domainuser1" MemberOf="CN=Remote Desktop Use
=Builtin,DC-peaksun,DC-org"
User SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=Domain User2,CN=Users,DC-peaksun,DC-org" Name="Domain
" ObjectGUID="abf722e0-2618-4243-b348-5cd95d37cca9" SamAccountName="domainuser2" MemberOf="CN=Remote Desktop Use
=Builtin,DC-peaksun,DC-org"
User SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=Domain Admin1,CN=Users,DC-peaksun,DC-org" Name="Domain
nl" ObjectGUID="4c0c30ee-a2e0-4d69-9c7d-6012f39bfc9" SamAccountName="domainadmin1" MemberOf="CN=Domain Admins,C
rs,DC-peaksun,DC-org, CN=Administrators,CN=Builtin,DC-peaksun,DC-org"
User SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=Rishi TestUser,CN=Users,DC-peaksun,DC-org" Name="Rishi
User" ObjectGUID="b6be867f-7670-473f-b729-07ac580948ba" SamAccountName="rishitestuser" MemberOf="CN=RishiTest2,C
rs,DC-peaksun,DC-org"
User SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=sumo,CN=Users,DC-peaksun,DC-org" Name="sumo" ObjectGUI
6c3d3fb1-5899-4269-a114-ab0c9b043a07" SamAccountName="sumo" MemberOf="CN=Engineering,OU=SIEM,OU=BusinessUnit1,DC-p
n,DC-org, CN=Remote Desktop Users,CN=Builtin,DC-peaksun,DC-org"
User SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=Rishi TestUser,OU=OU2,DC-peaksun,DC-org" Name="Rishi T
er" ObjectGUID="b73d84e1-54ba-414d-ae42-452b7c23a602" SamAccountName="rishitest3" MemberOf=""
User SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=dummy1,CN=Users,DC-peaksun,DC-org" Name="dummy1" Objec
="0b574bb6-639d-4793-894b-7554dce6a568" SamAccountName="dummy1" MemberOf="CN=Ductest1,CN=Users,DC-peaksun,DC-org
Group SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=Administrators,CN=Builtin,DC-peaksun,DC-org" Name="Ad
trators" ObjectGUID="16823247-b481-4dfe-8240-22e479f681ad" GroupCategory="Security" Members="CN=Domain Admin1,CN
s,DC-peaksun,DC-org, CN=Domain Admins,CN=Users,DC-peaksun,DC-org, CN=Enterprise Admins,CN=Users,DC-peaksun,DC-org,
ministrator,CN=Users,DC-peaksun,DC-org"
Group SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=Users,CN=Builtin,DC-peaksun,DC-org" Name="Users" Obje
D="cf164d13-3387-418b-b585-feac8c6f9f31" GroupCategory="Security" Members="CN=Domain Users,CN=Users,DC-peaksun,D
, CN=S-1-5-11,CN=ForeignSecurityPrincipals,DC-peaksun,DC-org, CN=S-1-5-4,CN=ForeignSecurityPrincipals,DC-peaksun,DC
Group SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=Guests,CN=Builtin,DC-peaksun,DC-org" Name="Guests" Ob
UID="2ce894e4-4bf0-48e6-a1d-ac5f94a017c0" GroupCategory="Security" Members="CN=Domain Guests,CN=Users,DC-peaksu
org, CN=Guest,CN=Users,DC-peaksun,DC-org"
Group SearchBase="DC-peaksun,DC-org" DistinguishedName="CN=Print Operators,CN=Builtin,DC-peaksun,DC-org" Name="P
```

To configure Script Sources:

1. Make sure a Collector is installed on a machine belonging to the domain managed by Active Directory.
2. Make sure that a Remote Windows Event Log Source is configured to collect events from each AD server.
(The sourceCategory should be set to OS/Windows.)
3. Create a PowerShell source for Script Source for domainCollector.ps1 and another Script Source for adObjectCollector.ps1:

Note: The **Frequency** option should be set according to your environment. We used Every 15 Minutes for our example, but in your deployment, the proper Frequency value depends on how often your topology changes. It's very important that the Frequency be set to a time longer than it takes for the script to run. For example, if a script takes two hours to finish, the Frequency should be set to Every 3 Hours. If the topology is relatively stable, the Frequency can be set to a longer value, such as Every 12 hours (it's recommend that each script run once every day).

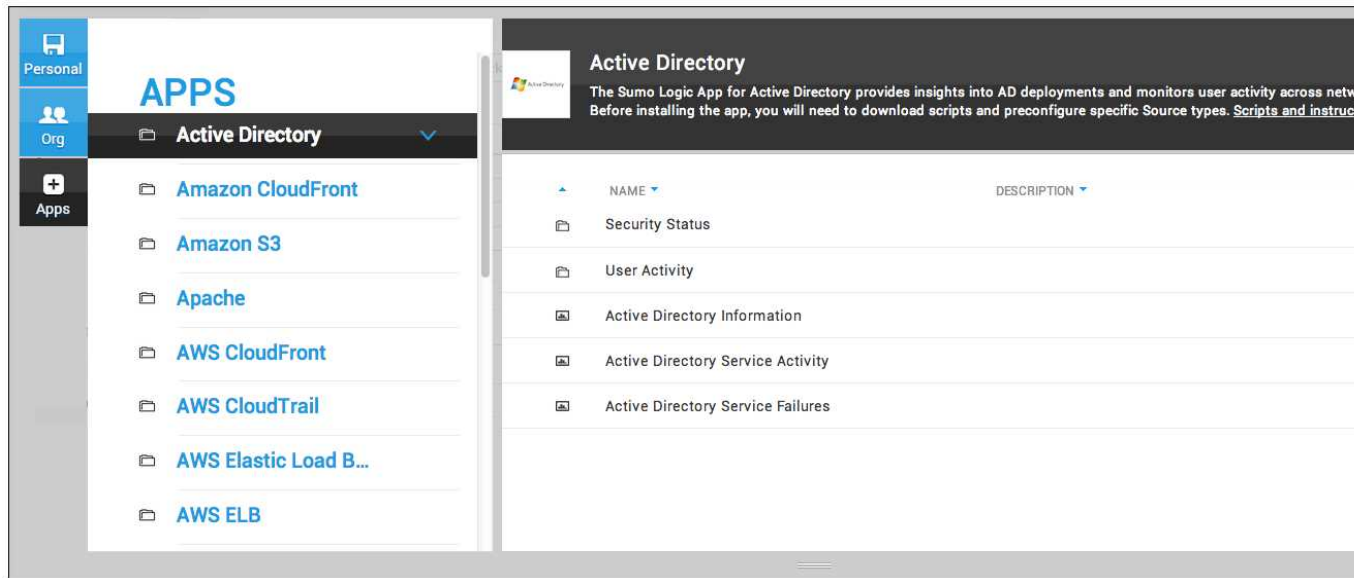
For more instructions on setting up a Script Source, see [Configuring a Script Source](#) in Sumo Logic Help.

Installing the AD App

After [configuring the collection](#) of Active Directory logs, you can install the Sumo Logic App for Active Directory to begin using the Dashboards and searches in the app.

To install the app:

1. In the **Library**, click the **Apps** tab.
2. Click **Active Directory**.
3. Click **Install**.



4. In the **Install Application** dialog box, select **AD/Topology** from the **_sourceCategory** menu. If you don't see this option, please make sure you've configured AD log collection properly. **Important:** If you do not select the correct **_sourceCategory**, data will not be loaded into the app.
5. For **Custom data filter**, type (**_sourceCategory=OS/Windows OR _sourceCategory=AD/Topology**).

6. Click Install.

Active Directory - Install Application

The Sumo Logic App for Active Directory provides insights into AD deployments and monitors user activity across network and... Before installing the app, you will need to download scripts and preconfigure specific Source types. [Scripts and instructions](#)

Data Source Mapping

Data Source Mapping determines what logs are included in the application Dashboards and searches. You may select a source category or define a custom keyword search that maps the application to your organization's data set.

[Custom data filter help](#)

☐ Select from `_sourceCategory` values

AD/Topology

Resulting data filter

`_sourceCategory=AD/Topology`

☒ Custom data filter

`(_sourceCategory=OS/Windows OR _sourceCategory=AD/Topology)`

Installation Location

Application Name*

Active Directory

Folder*

PERSONAL

Apache

Once an app is installed, it will appear in your **Personal** folder. From here, you can publish it to share it with your organization.

Monitors will start and fill automatically. It's important to note that each Monitor slowly fills with data as the length of the time range of each query is met. Results won't immediately be available, but with a bit of time you'll see full graphs and maps.

Sumo Logic App for Active Directory Dashboards

The Sumo Logic application for Windows Active Directory (AD) includes several Dashboards that allow you instant access to information about your system's visitors, traffic, and web server operations.

Active Directory Information



Topology. Displays your deployment's topology listing the forests, sites, domain DNs, and netbiosnames that have been active for the past two hours in a table.

Groups per Domain. Provides information on the distinct groups per domain in a bar chart for the past two hours.

Organizational Units per Domain. Shows the distinct organizational units per domain in a bar chart for the past two hours.

Computer OS per Domain. Displays the computer operating systems used by visitors to your site per domain for the past two hours.

Active Directory Service Activity



Top 10 Messages. Displays the top 10 messages reported in your system with message text and count in a table for the past 24 hours.

Rights Management. Reports the events for rights assigned or removed in timeslices of one hour for the past 24 hours using a combination line chart.

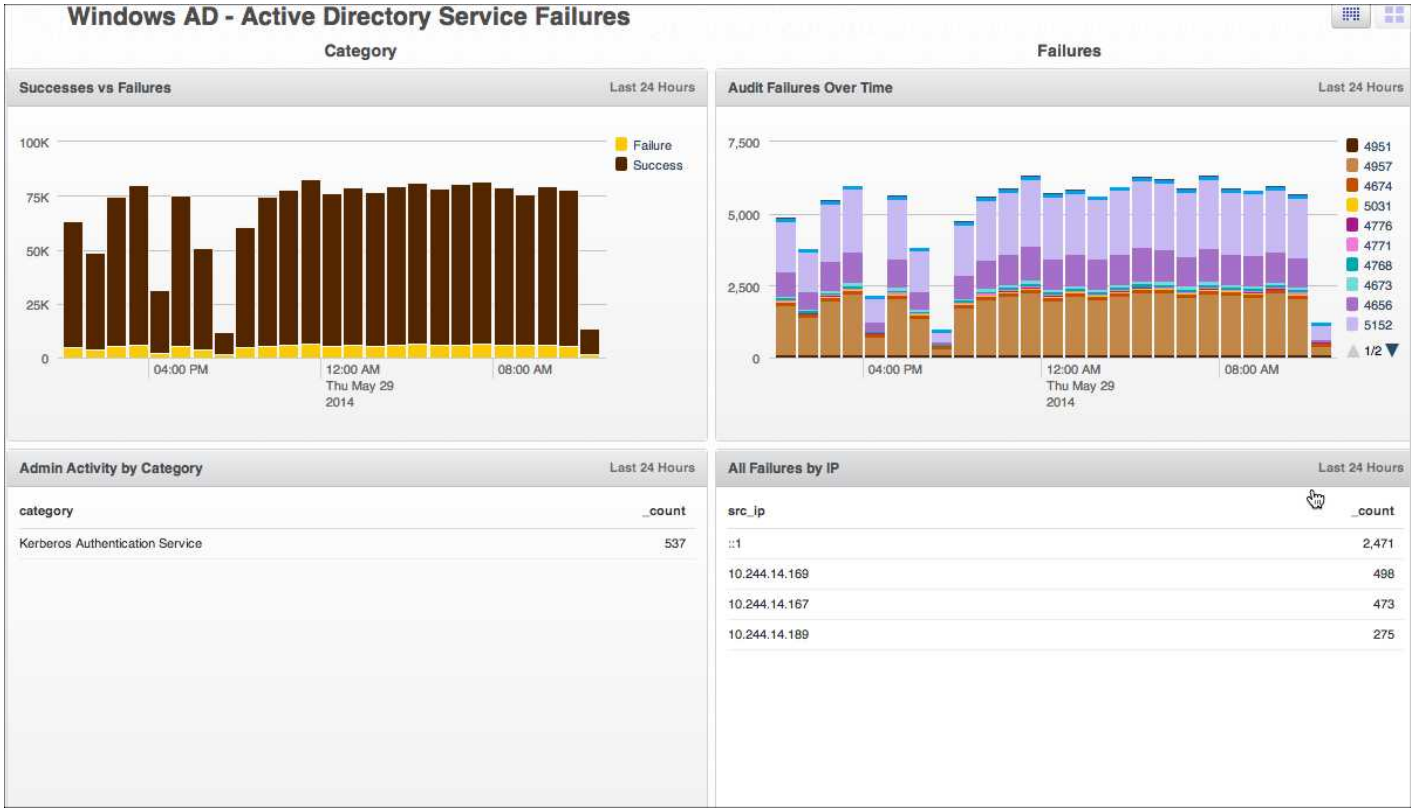
Messages Over Time by Category. Provides details on the messages reported by your system by category in timeslices of one hour over the last 24 hours, displayed in a combination line chart.

Logon/off Activity. Displays details on remote and interactive logon and logoff activity in timeslices of one hour for the past 24 hours using a stacked column chart.

Object Creation. Reports on creation events for users, computers, groups, and objects in timeslices of one hour for the past 24 hours using a stacked column chart.

Object Deletion. Reports on deletion events for users, computers, groups, and objects in timeslices of one hour for the past 24 hours using a combination line chart.

Active Directory Service Failures



- Successes vs Failures.** Displays the number of messages generated by your system for success vs failure in timeslices of one hour over the past 24 hours, in a stacked column chart.
- Admin Activity by Category.** Shows the administrator activity by category and count for the past 24 hours in a table.
- Audit Failures Over Time.** Displays the type and number of failures in timeslices of one hour for the past 24 hours in a stacked column chart.
- All Failures by IP.** Provides the IP addresses where failures have occurred along with the number of failures over the last 24 hours in a table.

Understanding Source Mapping in App Installation

When installing an app, choosing the correct sourceCategory and constructing the right data filter is very important. If you've installed an app and no data appears in an app's Dashboard, it could be due to an improper sourceCategory selection.

Apps are dependent on the sourceCategory metadata associated with your logs. This metadata is established when Collectors and Sources are configured. Because each organization uses a their own metadata methodology, you may need to check with your Sumo Logic account's Administrator to get a better idea of which sourceCategory would be the best fit for a given app.

How can I change the sourceCategory associated with an app?

To change the sourceCategory used by an app, you can simply reinstall the same app. (Currently apps cannot be uninstalled or edited through any other measure.)

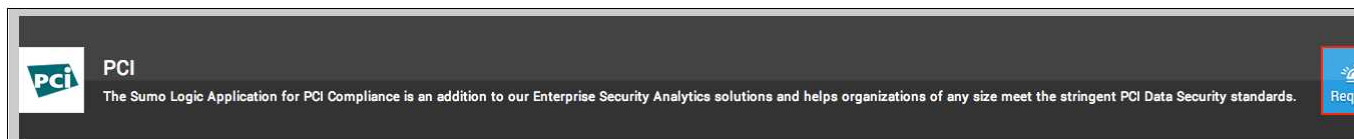
Requesting Apps

Although most apps can be installed directly from the Library, some apps are not available for instant installation. These apps may need custom setup, or may require a specific account type (such as the PCI app, which is available only to Enterprise customers). In these instances, an admin can request an app.

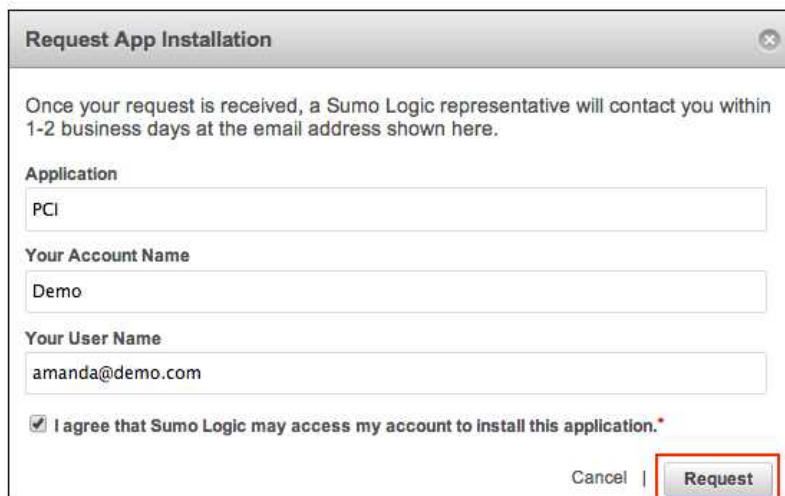
Once a request has been submitted, a support ticket is automatically opened. A representative from Sumo Logic will respond to the request as quickly as possible, generally between one and two business days. Depending on the app that's been requested, Sumo Logic may need additional information, or may need to work with your organization to change the account type to enable some apps.

To request an app:

1. In the **Library**, click the app you'd like.
2. Click **Request**.



3. Make sure the information in the **Request App Installation** dialog box is correct, then click the box to allow Sumo Logic to load the app in your account. Click **Request**.

A dialog box titled "Request App Installation" with a close button in the top right corner. The text inside says: "Once your request is received, a Sumo Logic representative will contact you within 1-2 business days at the email address shown here." Below this are three text input fields: "Application" with "PCI" entered, "Your Account Name" with "Demo" entered, and "Your User Name" with "amanda@demo.com" entered. At the bottom left is a checked checkbox with the text "I agree that Sumo Logic may access my account to install this application.*". At the bottom right are two buttons: "Cancel" and "Request", with the "Request" button highlighted by a red rectangle.

CHAPTER B

Sumo Logic Administration

Admins of Sumo Logic accounts have access to several Administration tools used to manage a deployment, including:

- Collector and Source management that allow Admins to upgrade Collectors and edit metadata.
- User and Role creation and administration.
- Security settings, including password and IP whitelist options.

Managing Installed Collectors and Sources

In the Collectors page of the Sumo Logic Web Application, you will see a list of all the Collectors and Sources in your current Sumo Logic deployment. You can manage your Collectors and Sources from this page.

In this section, you will learn how to:

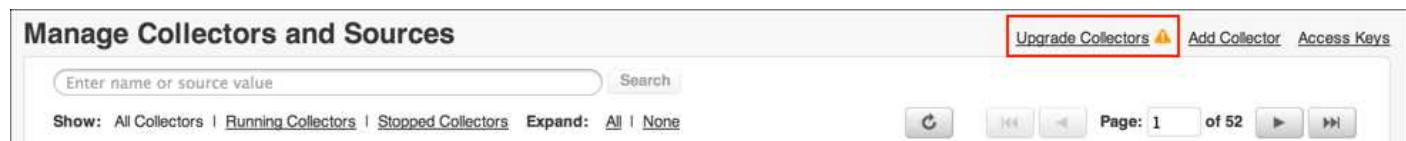
- [Search for a Collector or Source](#)
- [Add a new Source to a Collector](#)
- [Edit the name or metadata for a Collector](#)
- [Configure or edit a Local File Source](#)
- [Configure or edit a Remote File Source](#)
- [Configure or edit a Syslog Source](#)
- [Configure or edit a Windows Event Log Source](#)
- [Configure or edit a Script Source](#)

Upgrading Collectors using the Web Application

Deploying an upgrade to one or more Collectors can be done through the Sumo Logic Web Application. When you choose to upgrade a Collector, the upgrade file is automatically downloaded and executed from the Web Application instead of having to download the upgrade on each Collector manually.

How will I know when an upgrade is available?

When one or more Collectors in your Sumo Logic account are eligible for an upgrade you'll be notified through the **Collectors** page of the Sumo Logic Web Application:



Upgrading Collectors

You can choose to deploy the upgrade to all Collectors or you can pick and choose individual Collectors to install the upgrade (depending on scheduling needs or other factors).

To install an upgrade on one or more Collectors:

1. In the Sumo Logic Web Application select **Manage > Collectors**.
2. Click **Upgrade Collectors**.
3. Determine if you'd like to install the upgrade on individual Collectors or on all Collectors simultaneously. Then, choose one of the following:
 - Click **Update** next to the name of a Collector to install the upgrade just on that specific Collector. This option can be used if a policy prevents you from upgrading every Collector at the same time of day, or if you can't deploy the upgrade all at once. Any Collectors you choose not to upgrade will remain available in the Upgrade dialog box so you can install the upgrade at a later time.

- If you can safely upgrade all Collectors click **Update All**.

The upgrade process begins immediately after you click Update or Update All; the file is automatically downloaded and installed. You'll be notified when the upgrade has completed successfully.

Collectors that are offline or that have already been upgraded aren't eligible for upgrade and won't be included in the list of available Collectors in the Upgrade Collector dialog box.

Troubleshooting upgrade failures

If an upgrade fails, the red error icon appears in the dialog box, letting you know that the upgrade process was incomplete. Click **Retry** to start the process over.

The existing version will continue running on that Collector, with no disruption in service. However, it's important to keep Collectors up-to-date. If an upgrade fails, you'll receive a message indicating the problem, which may include any of the following:

- Available disk space on a Collector that wasn't successfully upgraded.
- Any permission errors that block access for Sumo Logic.
- Possible network failures that occurred during the upgrade.



If an upgrade repeatedly fails, [contact Sumo Logic Customer Support](#). Alternately, you can manually upgrade the Collector.

Searching for a Collector or Source

Many Sumo Logic customers have hundreds of Collectors and Sources installed and configured. But even with only 10 Collectors, sometimes it can be hard to find the one you need in the list.

On the **Manage Collectors and Sources** page, a search field allows you to search for a specific Collector or Source using complete keywords.



You can search on the following:

- Collector and Source name
- Collector and Source category
- Collector and Source type
- Collector and Source descriptions

To match partial keywords in a search, use a wildcard. For example, use **"apache*"** to match **"apacheprod"**.



Wildcards are supported only at the end of the keyword.

Search is case insensitive.

To search for a Collector or Source:

1. Go to **Manage > Collectors**.
2. Enter a complete keyword (or keyword and wildcard) in the search field, and click **Search** or **Enter**.

Search results are displayed.

To clear your search and display all Collectors again, refresh the browser.

Editing a Collector

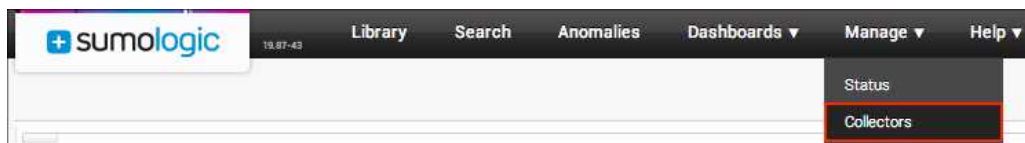
From the **Collectors** tab of the Sumo Logic Web Application, you can edit some characteristics of a Collector, including its name, description, Host Name, and Category.



Changes to metadata are applied to messages going forward from this point in time, and aren't applied retroactively.

To edit the name or metadata fields for a Collector:

1. In the Sumo Logic Web Application select **Manage > Collectors**.



2. Click the **Collector** name, or click the **Edit** link to the right of the Collector name.
3. Change the name or change the metadata fields as needed. If you set the Host Name or Category value at the Collector level, then all Sources belonging to this Collector are tagged with these metadata fields. If you later specify metadata at the Source level, the Collector metadata will be overwritten.

Edit Collector: A-test

Name * A-test

Version 19.0-344

Description

Host Name Test
Unless overwritten by Source metadata, the Collector will set the Source host name of all messages to this value.

Category test-nite
Unless overwritten by Source metadata, the Collector will set the Source category of all messages to this value.

Save | **Cancel**

4. Click **Save** to apply your changes.

Editing a Source

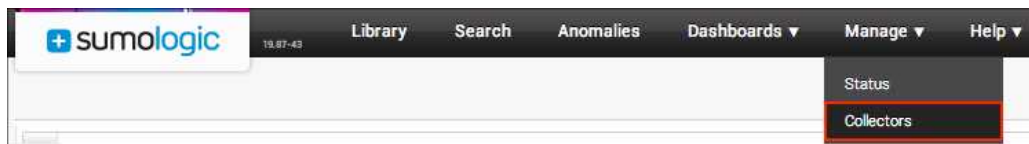
From the **Collectors** tab of the Sumo Logic Web Application, you can edit some characteristics of a Source, including its name, description, Collection time, Source Host, Source Category, Advanced options, and Filters.



Changes to metadata are applied to messages going forward from this point in time, and aren't applied retroactively.

To edit a Source:

1. In the Sumo Logic Web Application select **Manage > Collectors**.



2. Click the **Source** name, or click the **Edit** link to the right of the Source name.
3. Change the name or change the metadata fields as needed. (For details on configuration options, refer to your Source type in [Sources](#).)

Source Type Local File

Name* red5
Maximum name length is 128 characters.

Description

File Path* /Users/rosemary/ccn.log
Absolute path expression to one or more files,
or Windows UNC Share path for Windows collectors only.
For example: /var/log/messages or /var/log/*.log or \\hostname\\path\\to\\directory

Collection should begin All Time
(all historical information will be collected.)

Source Host Test
Host name for the local machine, e.g. LDAP_Server

Source Category AppName-test
Log category metadata to use later for querying, e.g. OS_Security

► Advanced

► Filters

Save | Cancel

4. Click **Save** to apply your changes.

Starting or stopping a Sumo Logic Collector

Sumo Logic Collectors start automatically at system startup. Manually restarting or stopping a Sumo Logic Collector requires root (Mac and Linux) or Administrator (Windows) privileges.

Linux/Solaris

If your underlying system supports **chkconfig** the Sumo Logic service is installed to start with system startup. To manually restart or stop the Sumo Logic Collector from the shell, log in as a root user. Go to the installation directory and type:

- \$./collector start
- \$./collector stop

Mac

The Sumo Logic Collector is installed to start at system startup. To manually restart or stop the Sumo Logic service, open a shell as root or as an authorized sudo user. Go to **/Applications/Sumo Logic Collector**, and type:

- \$./collector start
- \$./collector stop

Windows

The Sumo Logic Collector runs as a service and starts automatically at system startup. To manually restart or stop the Sumo Logic service:

1. Go to **Administrative Tools > Services**.
2. Select **sumo-collector** from the list.
3. Click **Stop** or **Restart**.

Filtering Source Data

Regular expressions are used to create rules that filter data sent to Sumo Logic from a Source. The filters affect only the data sent to Sumo Logic; logs on your end remain intact and unchanged. There are four different types of filters you can apply:

- **Exclude filters** are used to remove messages that you don't want to send to Sumo Logic at all (think of it as a "black list" filter). These expressions will be skipped.
- **Include filters** are used to send only the data you'd like in your Sumo Logic account (a "white list" filter). This type of filter can be very useful when the list of log data you want to send to Sumo Logic is easier to filter than setting up exclude filters for all of the types of messages you'd like to exclude.
- **Hash filters** replace an message with a unique, randomly-generated code to protect sensitive or proprietary information. You may want to hash unique identifiers, such as credit card numbers or user names. By hashing this type of data, you can still track it, even though it's fully hidden.
- **Mask filters** replace an expression with a mask string that you can customize—another option to protect data, such as passwords, that you wouldn't normally track.

How do Filters Work Together?

You can create any number of filters for a Source, combining the different types of filters to generate the exact data set you want sent to Sumo Logic. It's important to consider how filters work together:

- Exclude filters override all other filter types for a specific value. If you're excluding a value, it won't be sent to the Sumo Logic Cloud so it can't be hashed or masked.
- Mask and hash filters are applied after exclusion and inclusion filters to ensure that the inclusion filter sees log lines in their original state (rather than a log line with some values hidden).

Include and Exclude filters

You can use **include** and **exclude** filters to specify what kind of data is sent to the Sumo Logic Cloud. If you specifically exclude a message, it will never be sent to Sumo Logic. Think of an exclude filter as a blacklist filter.

Include filters are whitelist filters, which can be very useful when the list of log data you want to send to Sumo Logic is easier to filter than setting up exclude filters for all of the types of messages you'd like to exclude. For example, to include only messages coming from a Cisco ASA firewall, you could use the following:

Name	Filter	Type	Add Filter
Name	Cisco ASA firewall messages		
Filter	.%ASA-\d-\d{6}.*		
Type a regular expression that defines the messages you want to filter.			
Type	Include messages that match		
<input type="button" value="Apply"/> Cancel			

When writing your regular expression rules, keep in mind:

- Exclude rules always take precedence over include rules.
- If two or more rules are listed, the assumed Boolean operator is OR.
- The rule must match from the start to the end of any log message rather than addressing only a section. For example, if you want to exclude any message containing the words "secure" or "security", write the rule `.*secur.*`

Hash filters

With a **hash** filter, whatever expression you choose to hash will be replaced by a hash code generated for that value. Hashed data is completely hidden (obfuscated) before being sent to Sumo Logic. This can be very useful in situations where some type of data must not leave your premises (such as credit card numbers, social security, numbers, etc.). Each unique value will have a unique hash code.

For example, to hash member IDs, you could use the following:

member id	\[memberid=([^\]]*)\]	Mask	Edit	Delete
Name	member id			
Filter	\[memberid=([^\]]*)\]			
Type a regular expression that defines the messages you want to filter.				
Type	Hash messages that match			
<input type="button" value="Apply"/> Cancel				

Log line:

```
2012-05-16 09:43:39,607 -0700 DEBUG [hostId=prod-cass-raw-8]
```



```
[module=RAW] [logger=scala.raw.InboundRawProtocolHandler]
[memberid=dan@demo.com] [remote_ip=98.248.40.103]
[web_session=19zefhgy...] [session=80F1BD83AEBDF4FB]
[customer=00000000000000005] [call=InboundRawProtocol.getMessages]
```

Resulting hashed log line:

```
2012-05-16 09:43:39,607 -0700 DEBUG [hostId=prod-cass-raw-8]
[module=RAW] [logger=scala.raw.InboundRawProtocolHandler]
[memberid=3dfg3534ftgfe33ffrf3] [remote_ip=98.248.40.103]
[web_session=19zefhgy...] [session=80F1BD83AEBDF4FB]
[customer=00000000000000005] [call=InboundRawProtocol.getMessages]
```

Please note the following:

- Values that you want hashed must be expressed as a match group enclosed in "()".
- You can use an anchor to detect specific values. In addition, you can specify multiple match groups. If multiple match groups are specified, each of the values will be hashed uniquely:

Name	Filter	Type	Add Filter
Name	Filter	Type	
User email and IP address	(\b[A-Z09._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b)(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})	Hash messages that match	

Type a regular expression that defines the messages you want to filter.

Apply | Cancel

- If a match group isn't specified no data will be hashed.
- Make sure you don't specify a regular expression that matches a full log line. Doing so will result in the entire log line being hashed.

Mask filters

When you create a **mask** filter, whatever expression you choose to mask will be replaced with a mask string before it's sent to Sumo Logic (you can either select the character or use the default, #). For example, to mask all users' email addresses, you could use the following:

Name	Filter	Type	Add Filter
Name	Member email		
Filter	(\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b)		
	Type a regular expression that defines the messages you want to filter.		
Type	Mask messages that match		
	MASKED_USER_EMAIL		
	Type in a mask string, eg: #####		
	Apply		Cancel

Log line:

```
2012-05-16 09:43:39,607 -0700 DEBUG [hostId=prod-cass-raw-8]
[module=RAW] [logger=scala.raw.InboundRawProtocolHandler]
[auth=User:dan@demo.com] [remote_ip=98.248.40.103]
[web_session=19zefhgy...] [session=80F1BD83AEBDF4FB]
[customer=00000000000000005] [call=InboundRawProtocol.getMessages]
```

Resulting masked log line:

```
I2012-05-16 09:43:39,607 -0700 DEBUG [hostId=prod-cass-raw-8]
[module=RAW] [logger=scala.raw.InboundRawProtocolHandler]
[auth=User:MASKED_USER_EMAIL] [remote_ip=98.248.40.103]
[web_session=19zefhgy...] [session=80F1BD83AEBDF4FB]
[customer=00000000000000005] [call=InboundRawProtocol.getMessages]
```

Please note the following:

- Expressions that you want masked must be expressed as a match group enclosed in "()"
- You can use an anchor to detect specific values. For example, if in your logs all user emails can be identified in logs as `User: (user@email.com)` you could use `(User: ())` as an anchor.
- You can specify multiple match groups. Note that if multiple match groups are specified in one filter, each value will be masked the same way. So if you create one filter for users' email addresses and IP addresses

both will be replaced with the same mask string:

Name	Filter	Type	Add Filter
Name	User email and IP address		
Filter	<pre>(\b[A-Z09._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b) (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})</pre> <p>Type a regular expression that defines the messages you want to filter.</p>		
Type	Mask messages that match		
	USER_ADDRESS		
	Type in a mask string. eg: #####		
<div>Apply Cancel</div>			

- If you'd like to use a different mask for each value, you'll need to create a separate mask rule for each value. For example, if you'd like to mask IP addresses with a string that's different from the user email string, you'd create another filter with the expression `(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})` and you could use `USER_IP_ADDRESS` as the mask string:

Name	Filter	Type	Add Filter
Name	User IP address		
Filter	<pre>(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})</pre> <p>Type a regular expression that defines the messages you want to filter.</p>		
Type	Mask messages that match		
	USER_IP_ADDRESS		
	Type in a mask string. eg: #####		
<div>Apply Cancel</div>			

- Make sure you don't specify a regular expression that matches a full log line. Doing so will result in the entire log line being masked.

Creating filters

To create a filter for a Source:

1. Do one of the following:
 - To create a filter for an existing Source, in the Collectors page click **Edit** next to a Source. Click the menu next to **Filter**, and then click **Add Filter**.

- To create a new filter for a Source that you're configuring, click the menu next to **Filter**, and then click **Add Filter**.

▼ Filters

Name	Filter	Type	Add Filter

- To create a filter for an existing Source, in the Collectors page click **Edit** next to a Source. Click the menu next to **Filter**, and then click **Add Filter**.
- To create a new filter for a Source that you're configuring, click the menu next to **Filter**, and then click **Add Filter**.

▼ Filters

Name	Filter	Type	Add Filter

2. Type a **Name** for this filter.
3. For **Filter**, type a regular expression that defines the messages you want to filter. The rule must match the whole message. (This is as simple as adding `.*` to the front and back of the segment.)
4. Choose the **type** of filter you'd like to create:
 - **Exclude messages that match** to exclude the messages you've entered in the Filter text box.
 - **Include messages that match** to include the messages you've entered in the Filter text box.
 - **Hash messages that match** to replace the filtered data with a hash code.
 - **Mask messages that match** to replace the filtered data with a mask string entered in the text box under the Type menu. You can configure each mask filter to use a different character. If you don't specify a character, # will be used by default. For example:
5. Click **Apply**.
6. Click **Save** to return to the Collectors page.

Name	Filter	Type	Add Filter
Name	SSN		
Filter	([0-9]{3}-[0-9]{2}-[0-9]{4}) <small>Type a regular expression that defines the messages you want to filter.</small>		
Type	Mask messages that match MASKED_USER_SSN <small>Type in a mask string. eg: #####</small>		
<div> <div>Apply</div> <div>Cancel</div> </div>			

Editing filters

You can return to an existing filter at any time to make changes to the name, the filter expression, and the type of filter. Changes you make to a filter are not retroactive.

To edit a filter:

1. In the Collectors page click **Edit** next to a Source and click the menu next to **Filter**.
2. Edit any of the following options:
 - **Name** to change the name of the filter.
 - **Filter** to type a new expression.
 - **Type** to change the type of filter you run on this Source.
3. When you're done, click **Apply**.
4. Click **Save** to return to the Collectors page.

Establishing Metadata Conventions

Prior to installing Collectors, it is a good idea to establish predefined naming conventions for Sources, Collectors, and especially metadata tags. Sumo Logic Search supports the use of metadata tags in your messages such as Source Host and Source Category. This metadata is attached to your log messages at collection-time. All metadata is determined by the values you enter when configuring a Source. These tags are very important since they provide valuable keywords and terms you can use to find targeted results in search queries.

Suggestions for the following logical taxonomies are explained in the next sections:

- **Collector**. The name of the Collector entered at activation time.
- **Source**. The name of the Source entered when the Source is created.
- **Source Category**. This is a completely open tag determined by your entry to the "Category" field when you configure a Source. The tags you enter can help you to search by data type, machine type, function, location, or any category you choose without the need to specify which Collector or Source the messages belong to.
- **Source Host**. For Remote and Syslog Sources, this is a fixed value determined by the hostname you enter in the "Hostname" field (your actual system values for hosts). For a Local File Source, you can overwrite the host system value with a new value of your choice.
- **Source Name**. A fixed value determined by the path you enter in the "File" field when configuring a Source. This metadata tag cannot be changed.

Source Categories

Category metadata is a completely open metadata tag. The Source Category metadata is stored in a Sumo Logic field called `_sourceCategory`. This field is created when you enter text into the Source Category field at Source configuration time. If you prefer to set the tags higher up in the logical hierarchy, you can alternatively enter text in the Collector configuration for all Sources belonging to a Collector. For example, if you have three Syslog Sources feeding into one Collector, you might want to set a list of tags at the Collector level rather than tagging each Source separately. Note that the more specific Source-level tags override the more general Collector level tags.

Log categories can be somewhat complex, as many log files may belong to more than one logical category. For example, you may collect Apache logs for several reasons, including performance monitoring, security, and for audit compliance. Some of the major categories Sumo Logic recommends include:

- OS (for Operating System level logs)
- Security (for security related logs)
- Application (for application logs)
- Audit (logs for audit compliance)
- Performance (performance related logs)
- Debug (for application development debugging)
- Health (for system health logs)

In many cases, it may be difficult to foresee all the searches, reports and use cases you will eventually have for these log files. As such, Sumo Logic recommends chaining these metadata tags using underscores, and following a General-to-Specific order, like this:

- OS_Security
- OS_Audit
- OS_Application_Mail
- Security_Firewall_ASA
- Security_IDS_Snort
- Application_Apache_Performance
- Audit_SAS70
- Audit_Sorbox
- Debug_CustomApplication_Foo

The implied distinction between, for instance, OS_Application_Mail and say, Application_Mail would be for cases where you may simply be running the MTA (Mail Transfer Agent) that came with your flavor of Linux by default in order to send system notifications from cron jobs (OS_Application_Mail), versus, you are running an MTA as a service to provide email capabilities to your organization or customers (Application_Mail).

This allows you to do searches such as:

- `_sourceCategory=OS*`
- `_sourceCategory=*Audit`
- `_sourceCategory=*Application*`

Source Host

Hostname metadata is stored in a Sumo Logic field called `_sourceHost`. For the hostname, Sumo Logic retrieves and uses the host's actual OS-level hostname. For Local Sources, you can enter a different value in this field if you choose. If you choose to overwrite the system hostname, Sumo Logic recommends that you carefully select a meaningful name that uniquely identifies the host from which data is collected.

Remote collections present a special circumstance for Source Host metadata since one Remote Source can be configured to collect the same file from multiple hosts. In this case, Sumo Logic will tag each message with just one hostname (the host from which the message originated).

If you choose to overwrite the system names with custom metadata, the recommended best practice is to organize your hostnames in an easy to follow hierarchy such as:

- **Location_Purpose_UID**
- SF_MySQL_Primary
- Boston_FW_Secondary
- USEast_Hadoop_37

You will then be able to use wildcards to refine a search by any one of the chained terms in the Source Host metadata. For example:

- `_sourceHost=*USEast*`
- `_sourceHost=*MySQL*`
- `_sourceHost=*FW*`

Source Name

The metadata field called Source Name (`_sourceName`) contains the file path entered when you created your Source. If your Source points to more than one file path, then messages from each file path are tagged with the specific path from which they were collected.

Best Practice

When entering Source Metadata, it is strongly recommended that you use underscores to string together your metadata tags without using any space characters. Though spaces and special characters are allowed in the metadata fields, if you use spaces:

- You will need to quote the metadata exactly as you entered it at Source configuration time to find results.
- You will not be allowed to use wildcards since wildcards cannot be used with quotes.

If you use underscores instead of spaces, you will be able to use wildcards in your searches to match a partial tag for the metadata field. So, for example, we recommend you enter metadata into the Source Category field like this:

```
western_region_firewall_appliances_cisco
```

You can then search using wildcards by typing something like `_sourceCategory=*firewall*` in the query. Note that metadata tags can be changed later, but the changes are not retroactive. The new tags will only be applied going forward and cannot be changed for data that is already in the Sumo Logic Cloud.

Using the Status page

The Status page provides a visual snapshot of the overall message history for your deployment, as well as a message volume histogram for each Collector, giving you immediate visual feedback about traffic spikes or collection issues. To see statistics for any bar in the histogram, hover your mouse pointer over the area of interest.

When you first install a new Collector, after a few minutes of initial collection, the Status page shows a big spike of events in the message volume, and then fewer messages as the collection reaches a steady state. A local log file, for example, might contain millions of events. When the Collector is initialized, it quickly gathers all those entry logs and sends them to the Sumo Logic Cloud resulting in a traffic spike. After the initial collection, the Collector continues to tail the file (read from the end of the file as new entries are created) and send over the smaller number of new entries.

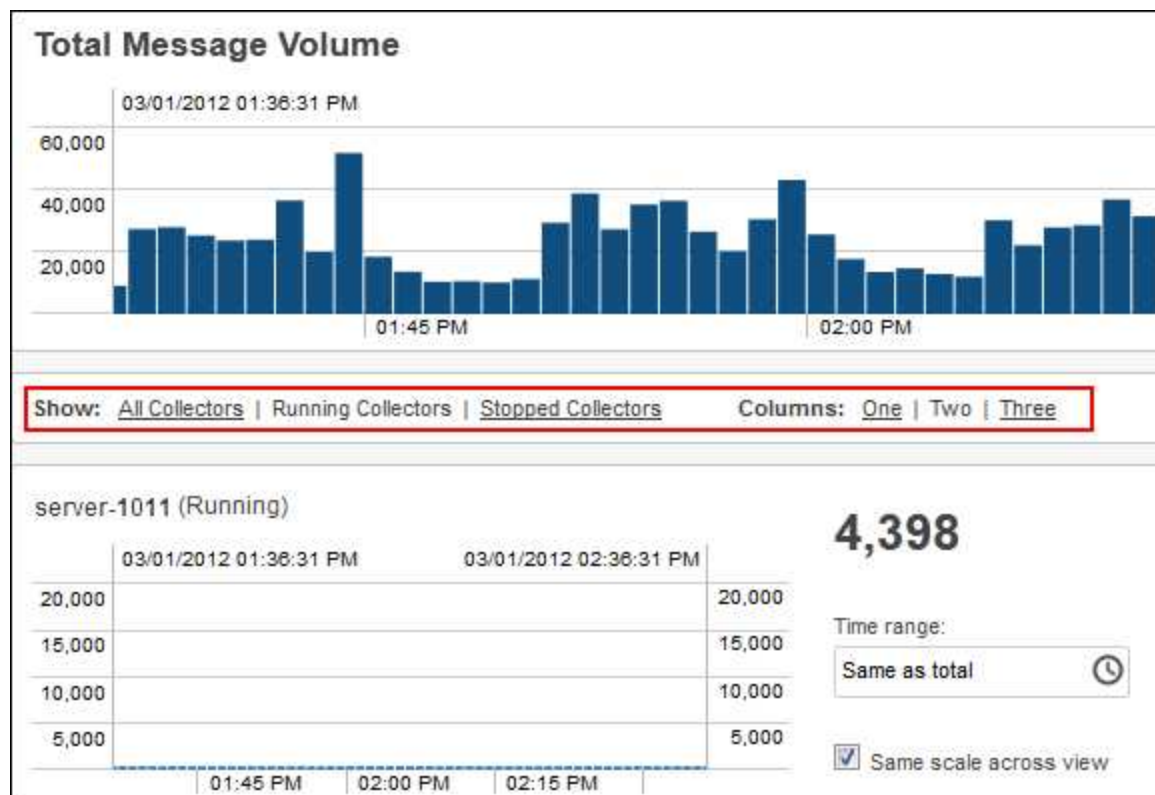
To view the Status page:

- Choose **Manage > Status**.

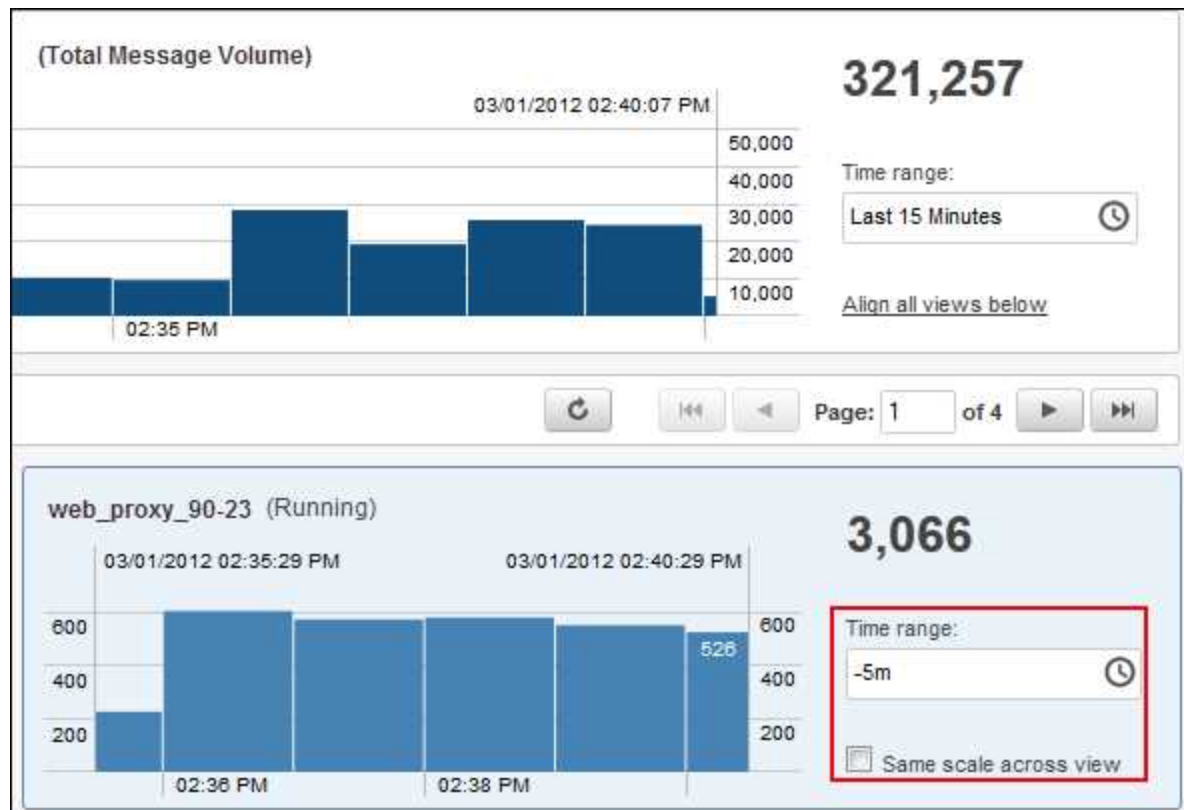
Changing the Scale or Timeframe for a Collector

In the Status page, you can:

- Choose to show all, only stopped, or only running Collectors.
- Display the Collector histograms in one, two, or three columns.
- Align the x-axis for each Collector to match the timeframe of the Total Message histogram.
- Scale the y-axis for each Collector to align with the other Collectors on the same page.



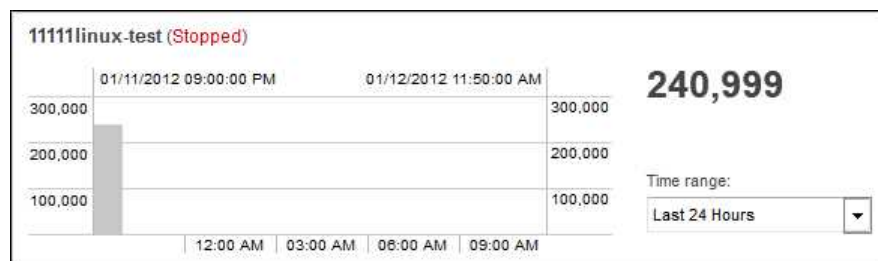
For each Collector, you can change the message volume scale so that variations in volume are easier to see. You can also change the timeframe for each Collector to investigate the stream volume for a single Collector. When a Collector's x or y axis is not aligned with all others, the background color changes to blue.



To return to an aligned scale across all Collectors, in the Total Message Volume area click the link to **Align all views below**. To return an individual view to the same scale as other Collectors, select the **Same scale across view** check box.

Viewing Stopped Collectors

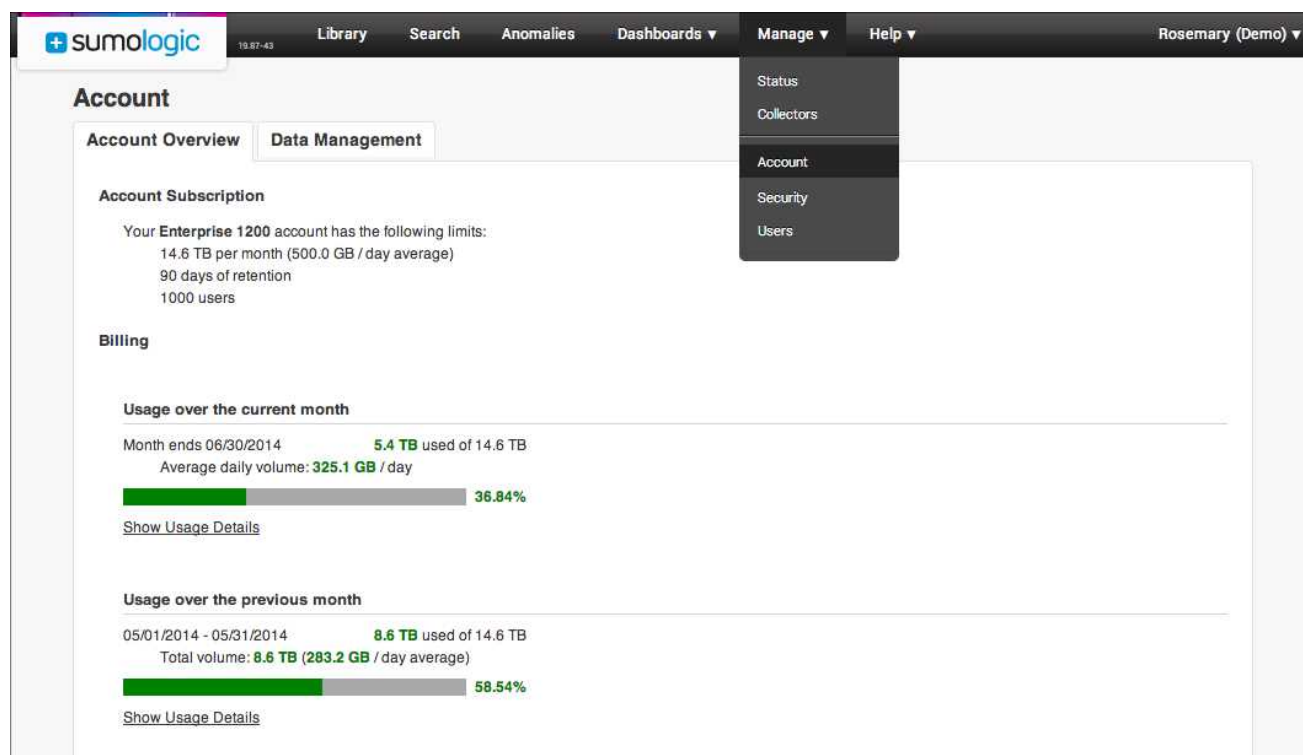
Stopped Collectors generally show zero messages in the traffic volume. Past messages display as gray bars in the histogram.



To make changes to the Collectors and Sources in your deployment, go to [the Collectors page](#). From there, you can start or stop Collectors, add new Sources, or edit Source configurations and metadata.

Account page

The **Account** page displays information about your Sumo Logic subscription, information on how much data you're uploading, and usage.



To view the Account page:

- Choose **Manage > Account**.

Account Overview

When you go to the **Account** page, the **Account Overview** tab is displayed. This page provides the following information about your subscription.

Account subscription

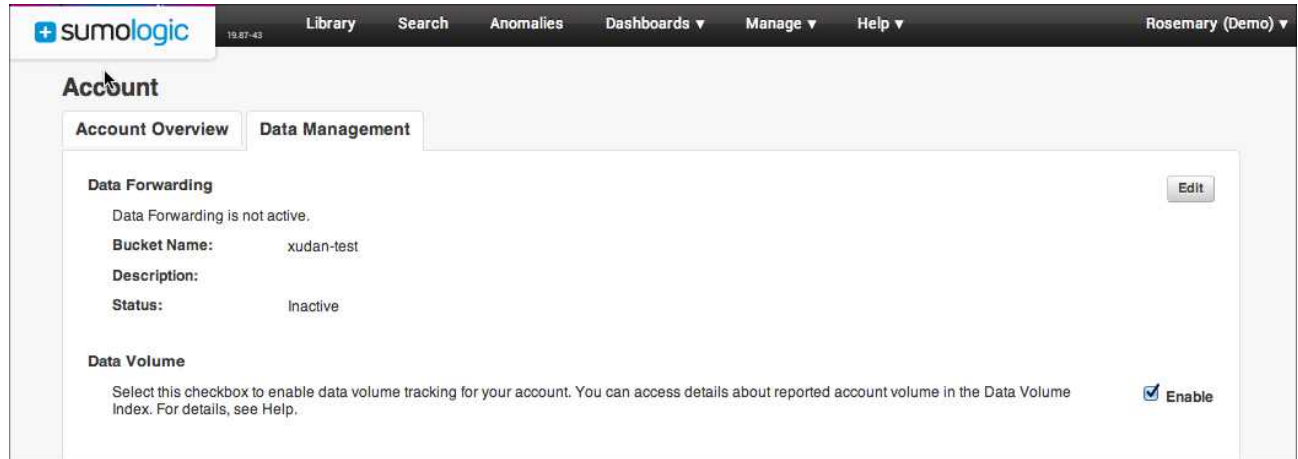
The **Account Subscription** section provides the details of your Sumo Logic subscription, including the data limits, retention limits, and the number of users allowed.

Billing

The **Billing** section provides usage graphs, which allow you to quickly view your available quota for the current billing period as well as for the previous billing period:

Data Management tab

The **Data Management** tab provides access to the Data Forwarding and Data Volume features.



Data Forwarding



This feature is only available to customers with Enterprise accounts, or those in paid trials. If you'd like to try Data Forwarding, please contact your sales representative.

When enabled, the Data Forwarding feature allows Sumo Logic to upload data to an S3 bucket that belongs to your organization. Log messages are saved as CSV files in compressed gzip files. They are accumulated and returned right after being ingested by Sumo Logic.

For more information, see [Data Forwarding](#).

Data Volume

The Sumo Logic Data Volume Index automatically provides data that allows you to understand your account's data ingest volume in bytes and number of log messages processed overall. Before it can be used, this feature must be enabled by an administrator.

For more information, see [Data Volume Index](#).

Data Forwarding

When enabled, the Data Forwarding feature allows Sumo Logic to upload data to an S3 bucket that belongs to your organization. Log messages are saved as CSV files in compressed gzip files. They are accumulated and returned right after being ingested by Sumo Logic.

In general, that means messages are flushed after 30 seconds, or when the total number of messages surpasses approximately 10,000 messages.

Data can only be sent to a single S3 bucket.

Also, please note that Sumo Logic can't monitor the amount of data being uploaded to the S3 bucket. You must monitor the size of the bucket, and follow any internal policies for managing data returned to your organization.

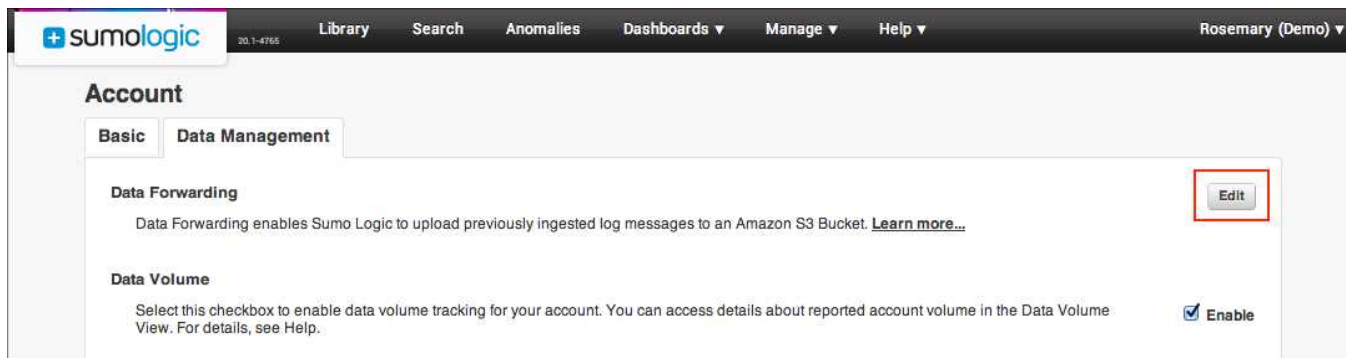
Before you begin

In order to configure Data Forwarding, you first need to make sure that you've given Sumo Logic privileges to access your organization's S3 bucket. This access is granted through Amazon Web Service Identity and Access Management (IAM). See [Granting Access to an S3 bucket for Data Forwarding](#) for more information and instructions.

Set up Data Forwarding

To set up Data Forwarding:

1. Make sure to have your AWS Key ID and Secret Key set up to allow Sumo Logic to write to the S3 bucket. You can learn more [here](#).
2. In the Sumo Logic Web Application, choose **Manage > Account**.
3. Select the **Data Management** tab.
4. To the right of **Data Forwarding**, click **Edit**.



5. Enter the exact name of the S3 bucket, along with the AWS Key ID and Secret Key generated for Sumo Logic. Make sure that **Active** is checked. Then click **Save**.

Data Forwarding

Bucket Name *

Description

Descriptive free-form notes

Key ID *

.....

Your AWS Access Key

Secret Key *

.....

Your AWS Secret Key

Active

☒ Enable or disable this Sumo Logic capability

Remove

Cancel

Save

After the credentials are verified by AWS, data will begin uploading to the S3 bucket.

Account

Account Overview

Data Management

✓ Your Data Forwarding configuration has been saved.

Data Forwarding

Data Forwarding is active and uploading to your AWS S3 bucket.

Bucket Name:

data-forward

Description:

Status:

Active

Editing Data Forwarding Settings

If you'd like to temporarily stop forwarding data, you can inactivate the S3 bucket. Additionally you can delete the existing bucket to permanently stop data forwarding, or to add a new bucket.

To cancel data forwarding:

1. In the Sumo Logic Web Application choose **Manage > Data Management**.
2. Click **Edit**.
3. Click **Remove**.
4. Click **Save**.

To delete S3 information:



Once the **Remove** button is clicked there is no way to undo this operation; the S3 information is immediately deleted.

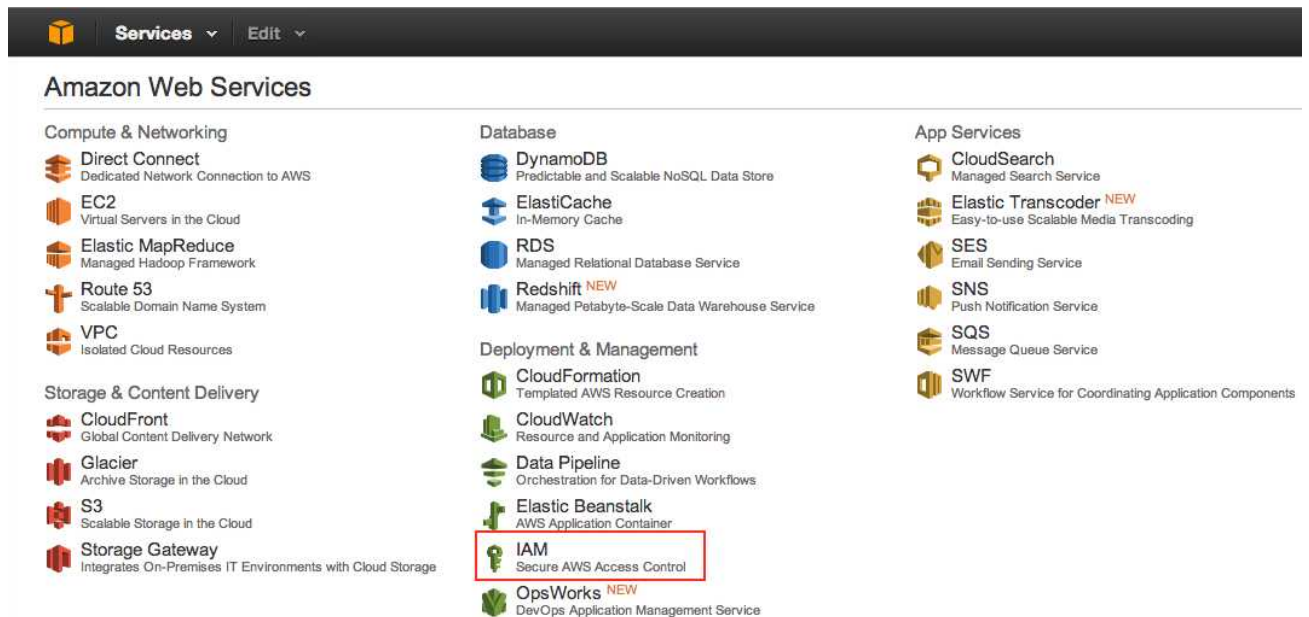
1. In the Sumo Logic Web Application choose **Manage > Data Management**.
2. Click **Edit**.
3. Click **Remove**.

Granting Access to an S3 bucket for Data Forwarding

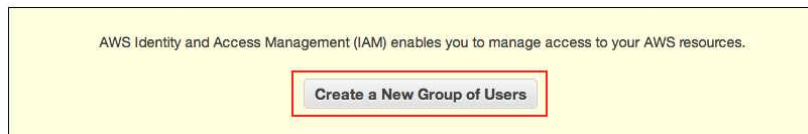
Before configuring Data Forwarding, you'll need to grant Sumo Logic permissions to access storage in your organization's bucket. First you'll need to make sure that your AWS account includes IAM. You can enable IAM through the AWS control panel.

Granting S3 permissions

1. In your AWS account Services page, click **IAM**.



2. Click **Create a New Group of Users**.



3. Create a group named SumoLogic, then click **Continue**.

Create a New Group of Users Cancel X

Group Name Permissions Users Review

Specify a group name. Group names can be edited any time.

Group Name:

Example: *Developers* or *ProjectAlpha*
Maximum 128 characters.

[Back](#) **Continue**

4. For Permissions, choose **Custom Policy**, then click **Select**.

Create a New Group of Users Cancel X

Group Name **Permissions** Users Review

Set Permissions

Select a policy template, generate a policy, or create a custom policy. A policy is a document that formally states one or more permissions. You can edit the policy on the following screen, or at a later time using the user, group, or role detail pages.

☐ **Select Policy Template**

☐ **Policy Generator**

☒ **Custom Policy**

Use the policy editor to customize your own set of permissions. **Select**

☐ **No Permissions**

5. For Policy Name, you may want to use **"put-s3-access"** or something similar, so your organization is aware of why this policy was created. Then, enter the JSON parameters you'd like to use for the policy (see this [JSON example](#) to copy and paste a recommended policy). Click **Continue**.

Create New Group Wizard Cancel X

Group Name Permissions Review

Set Permissions

You can customize permissions by editing the following policy document. For more information about the access policy language, see [Overview of Policies](#) in Using IAM. To test the effects of your policies before committing them into production, you can use the [IAM Policy Simulator](#).

Policy Name

put-s3-access

Policy Document

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "s3:PutObject" ],
      "Resource": [ "arn:aws:s3:::test/*" ]
    }
  ]
}
```

[Back](#) Continue

- Next, create a user (we named ours SumoLogic-data-forwarder) making sure to generate a key. Then click Continue.

Create User Cancel

Enter User Names:

1. SumoLogic-data-forwarder
- 2.
- 3.
- 4.
- 5.

Maximum 64 characters each

☒ **Generate an access key for each User**

Users need access keys to make secure REST or Query protocol requests to AWS service APIs.

For Users who need access to the AWS Management Console, create a password in the Users panel after completing this wizard.

Create

7. Click **Create**. The group is created.
8. Click **Show User Security Credentials** to view the **Access Key ID** and **Secret Access Key** for this user. You'll use it to set up Data Forwarding. You can also choose to download a .csv file with this information by clicking **Download Credentials**.

Policy JSON

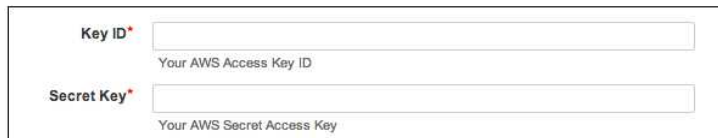
We recommend using the following JSON to create a policy:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "s3:PutObject" ],
      "Resource": [ "arn:aws:s3:::your_bucketname/*" ]
    }
  ]
}
```

Make sure to enter the actual name of your S3 bucket to the Resource line of JSON.

Managing Access Keys

In addition, while configuring an S3 Source, you'll need to provide **Key ID** and **Secret Key** credentials (tokens) to Sumo Logic. Security, token, and access settings are handled through **Amazon Web Service Identity and Access Management (IAM)**.



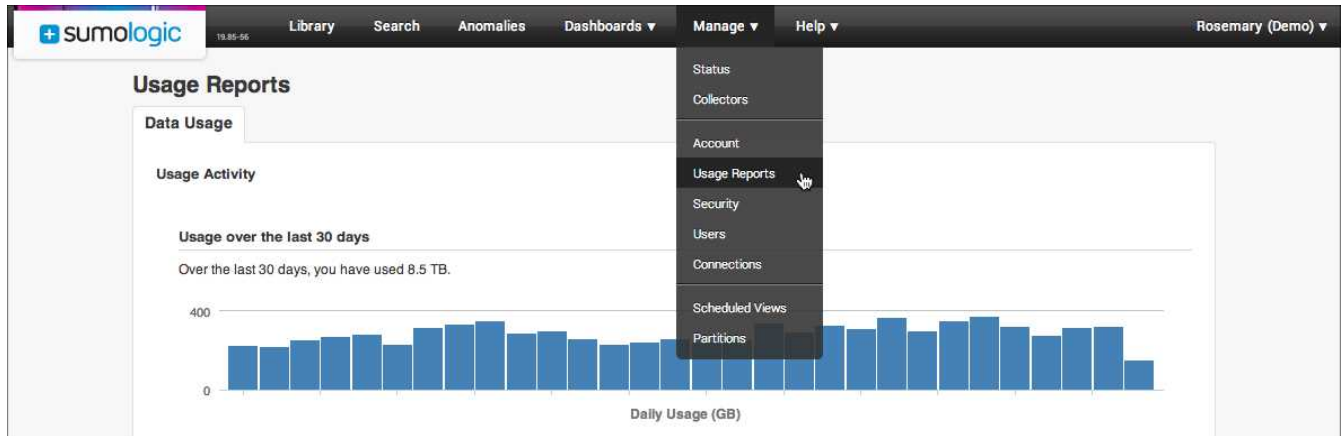
The image shows a form with two input fields. The first field is labeled 'Key ID*' and has a placeholder text 'Your AWS Access Key ID'. The second field is labeled 'Secret Key*' and has a placeholder text 'Your AWS Secret Access Key'.

Important: If your organization does not yet have IAM in your AWS account, you must add this option before configuring an S3 Source. Otherwise Sumo Logic won't have appropriate permissions to access your data.

For instructions on using IAM, please see [AWS Identity and Access Management \(IAM\)](#) to learn about the options available to your organization.

Managing Usage Reports

The **Usage Reports** page displays important information about your account's data usage.

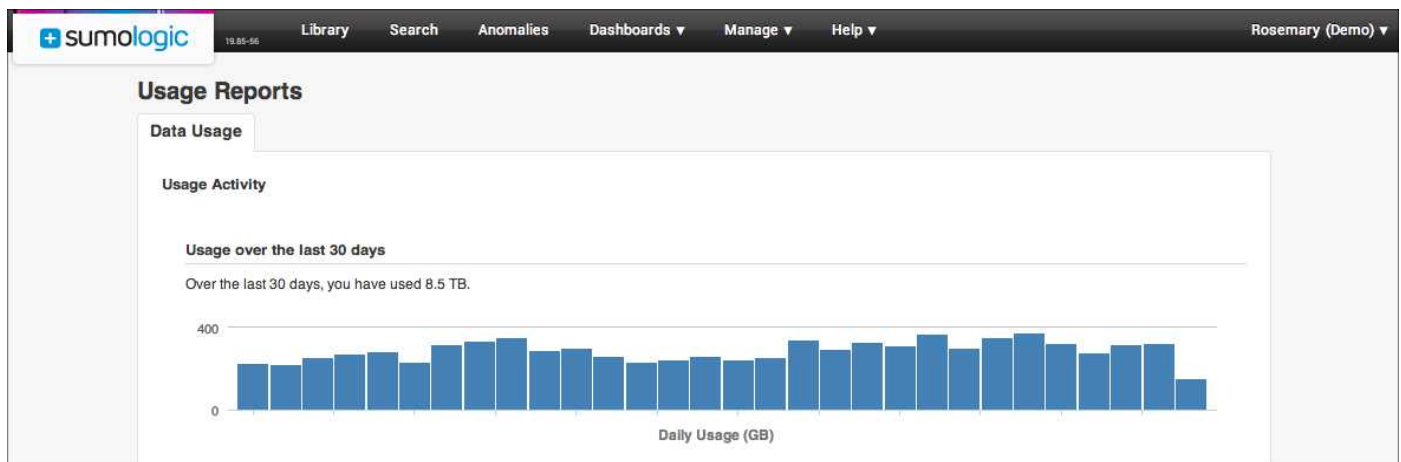


To view the Usage Reports page:

- Choose **Manage > Usage Reports**.

Data Usage

The **Data Usage** tab is the first tab on the **Manage > Usage Reports** page. It provides details about your account's usage activity.



Usage over the last 30 days

At a glance, this graph lets you view the usage activity for your account in MBytes by day for the last 30 days.

Managing Data Volume

With Sumo Logic Free and, depending on your subscription, Sumo Logic Professional accounts, daily data volume is limited. It's important to keep track of your daily usage to avoid overages, and to know when you may need to upgrade your account. Here's information on monitoring and limiting the data you're sending to the Sumo Logic Cloud.

Monitor average daily volume

You can see the average amount of data you're sending to the Sumo Logic Cloud on the **Account** page. If you notice a trend towards going past the daily limit on your account, you can take actions to reduce your usage (see the next section) or contact Sumo Logic to upgrade your account.

Use filters to limit the data volume

Using an exclude or include filter can help you limit the log messages you're sending. Depending on the situation, you could do one of the following:

- If there are certain types of messages that you don't need in your account, create [exclude filters](#) to prevent that data from being uploaded to the Sumo Logic Cloud.
- If you know exactly what messages you'd like to send to Sumo Logic, create an [include filter](#) to specify those messages.
- If you've noticed a big spike in data usage on a Source or Collector, you could elect to take it offline temporarily in order to remain under the usage limits.

If you have questions or need more information, please contact [Sumo Logic Support](#).

Using Role-Based Access Control with Sumo Logic

Sumo Logic now supports Role-Based Access Control (RBAC) to allow Administrators to customize system access. With RBAC, Administrators create roles that are created for groups of users who perform various job functions. Users are not assigned permissions directly, but inherit permissions through roles (or even through a single role). Role assignments can grant users permissions to access some data sets, or can restrict users from accessing types of data.

These benefits extend beyond IT or operations functions. For example, let's say we're designing a role for a sales team. The sales team needs a very targeted subset of data to see who is accessing a portal, giving them insights that they can use to go after leads or cultivate prospects. Or, for a completely different example, in a hospital setting, data related to patients' personal information uploaded from a specific Collector can be completely segregated from other data types, making sure that both security and patient confidentiality policies are met simultaneously.

Administrators benefit from a more streamlined user management process with RBAC. Instead of creating permissions based on a user, and then having to repeat the process with every user, Admins can simply assign roles to a user's account. For example, an IT group can quickly be assigned a role that sets access and permissions that should be identical across the team. You'll set up the role once, then assign it to each user with just a few clicks. Roles can be created on a per-group basis, per-job function basis, and so on.

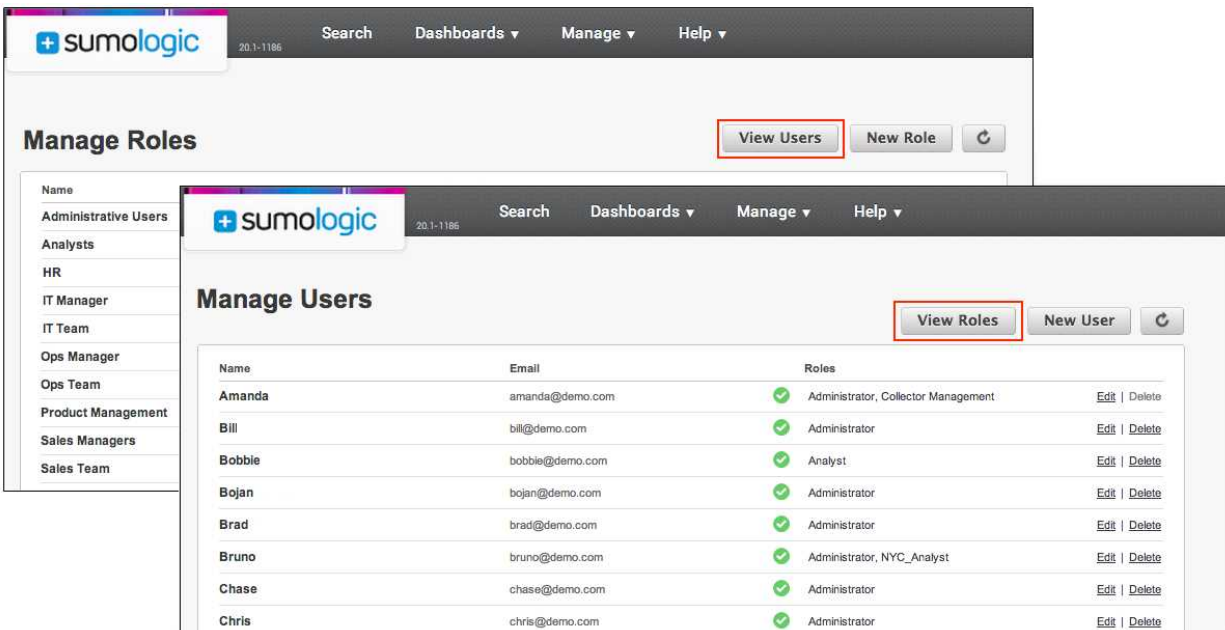
RBAC can also help with site-specific data. Perhaps an Admin would like to assign roles to users geographically, where a group in one location is managed through one role, while a similar group at another site is assigned a different role—each group has access to site-specific data, allowing them the permissions they need to perform their duties, while preventing sensitive information from leaving a site.

In addition to data access, user roles are also used to grant permissions to Collectors, restricting access to Collector management (which refers to installing, upgrading, and monitoring Collectors) to only the users who need that level of control over the operations of a Sumo Logic account.

Managing Users and Roles

Tools used to manage users and roles are located in the Sumo Logic Web Application. Two pages toggle between user management options and role management options.

All user creation and editing is available via the **Manage Users** page, while all role creation and editing is done through the **Manage Roles** page. To toggle between the pages you'll click the **View** buttons in the top right of each page:



To see the **Manage Roles** page, click **View Roles** in the Manage Users page. To see the **Manage Users** page, click **View Users** from the Manage Roles page.

Managing User Roles

Administrators of Sumo Logic accounts are able to create, edit, and delete roles to maintain compliance with internal data access policies.

In addition to two default roles, any number of customized roles can be created and assigned to users, to either limit or grant access to data and Collectors.

Understanding default user roles

There are two default roles assigned to users in a Sumo Logic account: **Administrator** and **Analyst**. These roles are assigned to any users already in your organization's Sumo Logic account.

If your organization isn't ready to implement RBAC, you can keep users assigned to these two roles, just as before. By default, both roles have full access to data in an account.

Administrator

Because the Administrator role is designed as a "super-user", the Administrator role cannot be deleted. It's created by default with full permissions and rights.

Only Administrators can:

- Manage users (create new users, deactivate users, re-activate users, and delete users).
- See a list of all roles in the account.

- Create new roles (manage access to data).
- Apply roles to a user.
- Edit roles.
- Delete roles.

In addition, Administrators have permissions to read and manage Collectors, meaning that in addition to running queries, Administrators can download and install Collectors, upgrade Collectors, and view information on the **Collectors** page of the Sumo Logic Web Application.

Analyst

Users within the Analyst role have permissions to access every part of a Sumo Logic account except for user and Collector management. This role can be deleted if it doesn't suit your organization's needs (after you've reassigned all users to other roles).

By default, Analysts have access to read Collectors and Sources, and are able to run queries on all the data your organization has uploaded to Sumo Logic. You can, however, change the Analyst privileges and permissions if necessary.

Defining custom roles

Roles are specified using search language constructs in a Query String. The string is added silently to every query run by a user assigned to a role.

In addition, roles determine who has the ability to manage Collectors in a Sumo Logic account.

When creating a new role, you'll need to specify the following:

- **Name.** Choose a name that concisely describes the role.
- **Description.** An optional description of the role.
- **Query String.** Determines the data users assigned to the role can access. For more information, see [Constructing a query string](#).
- **Permissions.** Determines a role's ability to read or manage Collectors. Roles that are set to only Read Collectors can run searches, but cannot install or upgrade Collectors; the Manage Collectors option allows a user to install, upgrade, and otherwise manage all the Collectors (and Sources) in a Sumo Logic account.

How do roles work together?

When a user is assigned to more than one role, the rules are combined with an **OR** in front of each query the user runs. Users inherit the highest level of access granted in the roles they are assigned. For example, if a user is assigned to the role "admin" which has the most capabilities, and also to a role "advanced user" which a different set of capabilities, the user will have the capabilities of both roles.

For example, let's say an IT team member is assigned to roles that, when combined, allows him to access to a Collector named firewall, yet no access to a Collector named HR. His combined query string (added to every search) would look like:

```
_collector=firewall OR !_collector=hr AND [search query]
```

He's able to see IT-related errors and activity, but HR records remain out of his search results.



To restrict a user to a certain level you'll need to remove that user from the Administrator role and add him or her to a role with fewer permissions.

What about shared content?

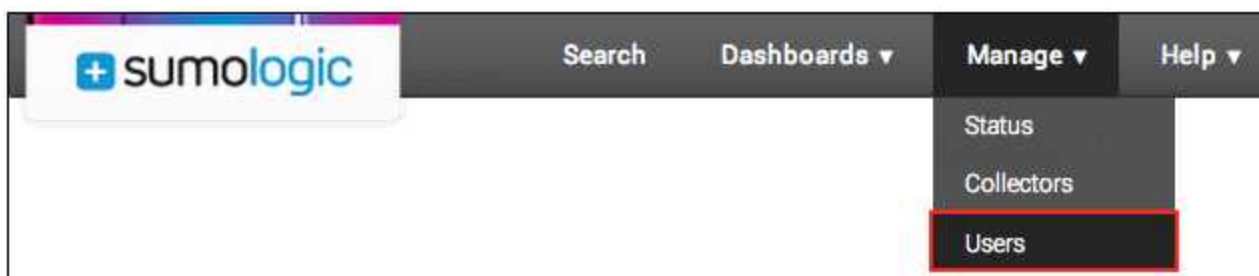
The results of shared searches will be filtered by the role of the person who shared the search). For example, if a user assigned to roles that allow a higher level of access shares a Dashboard, users with lower levels of access will see the same data—the Monitor is run against the user's permissions, and the resulting data displays.

The same is true of shared searches. If a user with a low level of access shares a search, when a user with higher access permissions runs the query he or she will see a different set of results.

Creating a new role

New roles are created in the Manage Roles page of the Users tab.

1. Click **Manage > Users** in the Sumo Logic Web Application.



2. Do one of the following:
 - If the **Manage Users** page is displayed, click **View Roles** (to display the Manage Roles page), then click **New Role**.
 - If the **Manage Roles** page is displayed, click **New Role**.
3. Enter the following:
 - **Name.** Choose a name that concisely describes the role.
 - **Description.** An optional description of the role to help other Administrators understand the purpose or limitations of the role.
 - **Query String.** Type the string you would like to add as a prefix to every query run by users assigned to the role. For more information see [Constructing a query string](#).
 - **Permissions.** Select **Read Collectors** or **Manage Collectors**. Roles that are set to **Read Collectors** can run searches, but cannot install or upgrade Collectors; the **Manage Collectors** option allows a user to install, upgrade, and otherwise manage all the Collectors in a Sumo Logic account.



All users granted Manage Collectors permissions are able to access machines running Collectors.

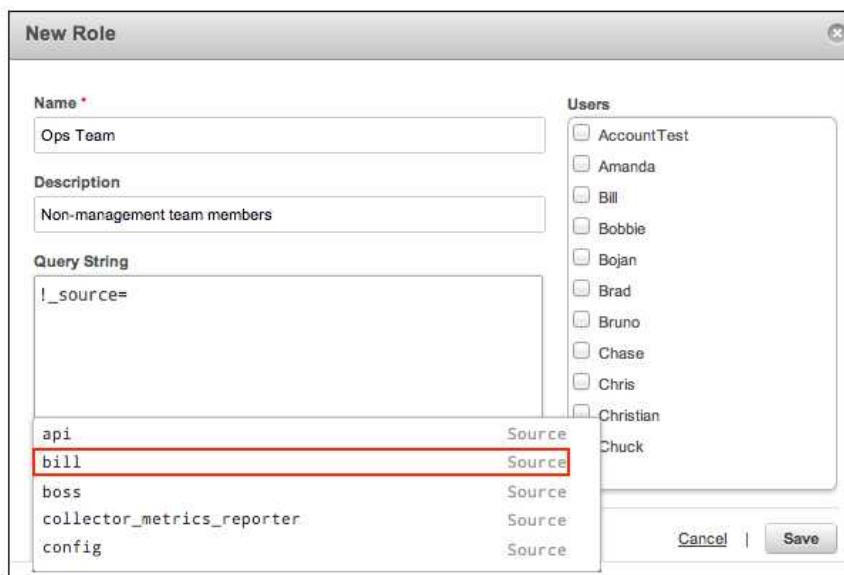
4. In the **Users** list, click the check box to the left to the name of each user you'd like assigned to the role.
5. Click **Save**.

Constructing a query string

A Query String defines access for a role. Limitations can be set based on several types of metadata, or at the record level. String are silently added to the beginning of each query a user runs.

As you type a string in the **New Role** dialog box, supported metadata and record options are displayed to help you enter a well-defined string. Any errors in the string are underlined in red.

For example, when creating the Ops Team Role, we first want to prevent Ops Team members from viewing billing information, which comes from a source named **bill**. When we type `!_source` into the Query String box, the list of Sources in our account appears, allowing us to see what options are allowed. We can either click the Source, or continue typing to see more options.



Role strings cannot include vertical pipes, "|".

Denying access to data

Denying a group of users to data is as simple as typing `!"` in front of the metadata object or field that a role should not access. The `!"` is treated as NOT in Sumo Logic queries.

Using our example from above, when a user assigned to the Ops Team role runs a search, `!_source=bill AND` is added to the search. So, for example, say that an Admin user ran a query, like:

```
(error OR fail*) AND exception | count by _sourceCategory| sort by _count
```

The query produced aggregate results by sourceCategory—the results shown below on the left. If an Ops Team member ran that same query, he or she would see the results shown below on the right, because the search that is run looks like:

```
!_source=bill AND(error OR fail*) AND exception | count by _sourceCategory| sort by _count
```

Note that bill isn't represented at all because of the prefix.

Messages		
Aggregates		
Page: 1 of 1		
#	_sourcecategory	_count
1	search	2,767
2	nova	965
3	bill	828
4	api	528
5	service	371
6	raw	357
7	config	320
8	nerv	229

Messages		
Aggregates		
Page: 1 of 1		
#	_sourcecategory	_count
1	search	2,767
2	nova	965
3	api	528
4	service	371
5	raw	357
6	config	320
7	nerv	229

Granting access to data

What if you'd like to limit a role to a specific data set, or if you have a Collector that should only be accessed by certain users? Instead of denying access, you'll grant access to a data set by just typing that object in the Query String text box.

For example, say we'd like to create a role for a team that needs to monitor firewall logs that are received by a specific Collector. You could construct a query string like `_collector=firewall*`, which adds `_collector=firewall* AND` to each query run by users in that group.

Defining access based on metadata

You can restrict access to the following metadata fields:

- `_collector`
- `_source`
- `_sourceCategory`
- `_sourceHost`
- `_sourceName`

For example, let's say that our admin wants to create a role that prevents access to hosts named **humanresources**, **finance**, and **secret**. To create this role, he would type the following string:

```
!_sourceHost=humanresources* and !_sourceName=*finance* and !_sourceCategory=*secret*
```

If a user with the above role runs a query like:

```
error | count by _sourceHost
```

the following query is what Sumo Logic actually runs:

```
error and (!_sourceHost=humanresources* AND !_sourceName=*finance* AND !_sourceCategory=*secret* | count by _sourceHost
```

The results will exclude all results that were described by the role's restriction.



For more information on Sumo Logic metadata fields, see [Metadata Searches](#).

Defining access based on records

You can create roles based on removing access to specific values in log messages. For example, let's say you'd like to create a role for a subset of users that should never see data that contain `userid`, or anything that matches `secret*`.

In this case, constructing a role using this string:

```
!userid and !secret*
```

means that if a user with the above role runs a query like:

```
error | count by _sourceHost
```

the following query is what Sumo Logic actually runs:

```
error AND(!userid AND !secret*) | count by _sourceHost
```

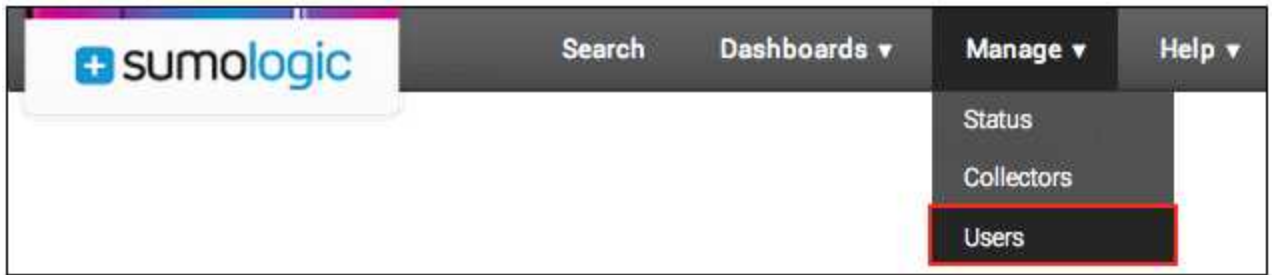
Assigning roles to users

Before you implement RBAC, all of the users in your account are assigned either the Administrator or Analyst default role (depending on which role they had previously).

When you're ready to assign roles, you'll need to go through and edit settings for anyone that should not be an Administrator. Remember, users inherit the highest level of access in their assigned roles, so as long as a user is considered an Administrator he or she has full access to all data and Collectors in your Sumo Logic account.

To assign a role to users:

1. Click **Manage > Users** in the Sumo Logic Web Application.



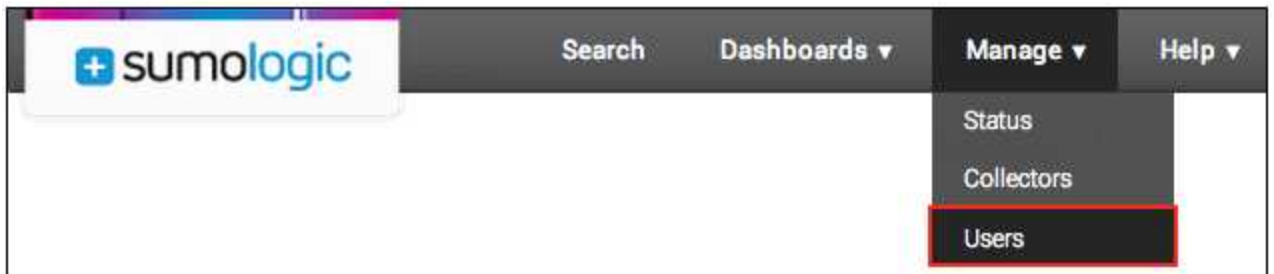
2. If the **Manage Users** page is displayed, click **View Roles**.
3. Click **Edit** at the far right end of the role's line.
4. To assign the role to users, click the check box to the left of each user's name.
5. When you're done, click **Save**.

Editing or deleting roles

Roles can be edited at any time. Query strings can be modified, and Collector Permissions can be changed. You can also change the users assigned to a role.

To edit a role:

1. Click **Manage > Users** in the Sumo Logic Web Application.



2. If the **Manage Users** page is displayed, click **View Roles**.
3. Click **Edit** at the far right end of the role's line.
4. Change any of the following:
 - **Name.** Type a new name for the role.
 - **Description.** Type or modify the description.
 - **Query String.** Edit the limitations of the role.
 - **Permissions.** Select or deselect **Read Collectors** or **Manage Collectors** to modify those options.
Roles that are set to only Read Collectors can run searches, but cannot install or upgrade Collectors; the Manage Collectors option allows a user to install, upgrade, and otherwise manage all the Collectors in a Sumo Logic account.
5. Click **Save** when you're done editing.

Roles that are assigned to one or more users cannot be deleted.

To delete a role:

1. Click the **Users** tab in the Sumo Logic Web Application.
2. If the **Manage Users** page is displayed, click **View Roles**.
3. Click **Delete** at the far right end of the role's line.
4. When asked to confirm the deletion click **OK**.

Managing Users

Creating new users, as well as editing the roles and credentials of existing users, is managed through the Sumo Logic Web Application.

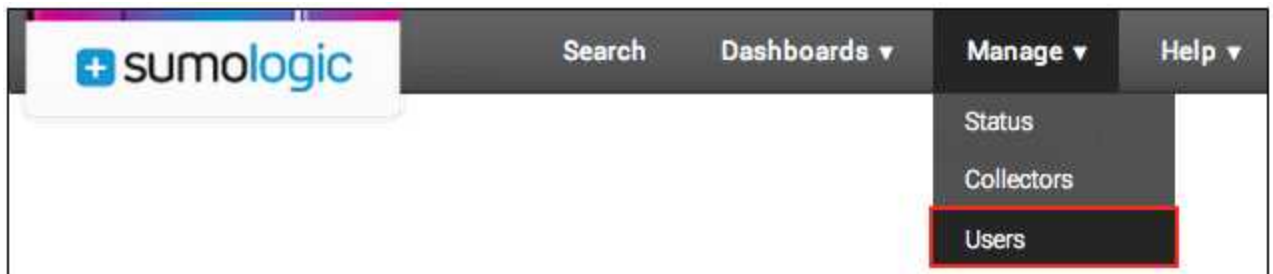
In addition, Administrators can reset an existing user's password. Note that all reset passwords (as well as initial passwords for new users) are generated by Sumo Logic; when the user logs in for the first time (or after a password reset for existing users) the user is prompted to change the temporary password before the log in is completed.

Creating a new user

When you create a new user, you'll need to make sure to assign at least one role to the user; no default roles are given to a new user.

To create a new user:

1. Click **Manage > Users** in the Sumo Logic Web Application.



2. Do one of the following:
 - If the **Manage Roles** page is displayed, click **View Users** (to display the Manage Users page), then click **New User**.
 - If the **Manage Users** page is displayed, click **New User**.
3. Enter the user's credentials:
 - **First Name** and **Last Name**. Enter the first and last name of the user.
 - **Email**. The email address must match the corporate domain of your Sumo Logic account, and cannot be edited later. For new users, Sumo Logic automatically sends a temporary password to the user at the email address you've entered. The first time a user log in to Sumo Logic, he or she is prompted immediately to change the temporary password.

- **Status.** By default a new user's status is set to **Active**. However, you might choose **Inactive** for a user who has not yet started work at the company, or to prevent access for an existing user. This setting can be edited at any time. Users are not notified of Status changes.
4. For **Roles**, select the roles you'd like to assign the user:

5. Click **Save**. The user is immediately added to your account.

Editing user credentials or roles

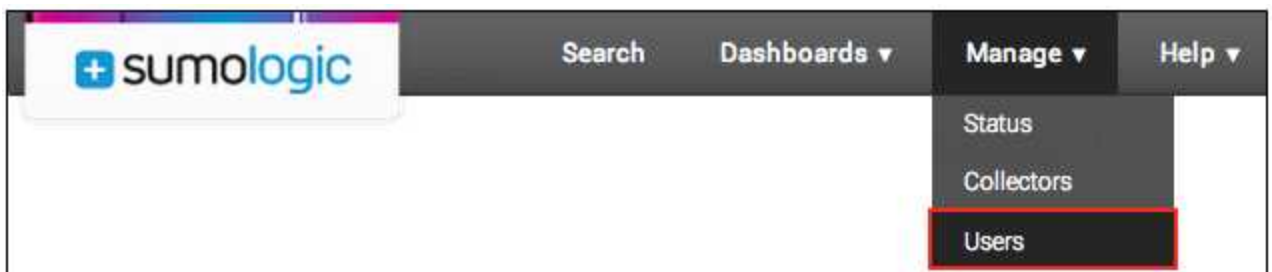
When you add or remove a role from a user the change is implemented right away, granting the user the highest level of access of the combined roles.

Editing a user

In addition to changing the roles assigned to a user you can edit a user's name and Status. However, the email address associated with a user cannot be edited; if a user's email address changes you'll need to create a new user and then delete the existing account.

To edit a user:

1. Click **Manage > Users** in the Sumo Logic Web Application.



2. Make sure the **Manage Users** page is displayed. (If the **Manage Roles** page is displayed, click **View Users**.)

3. Click **Edit** to the right of the user's name.
4. Edit any of the following:
 - **First Name or Last Name.** Make any changes to the user's name.
 - **Roles.** To remove a user from a role, deselect the role; to add the user to a role, select the role.
 - **Status.** Select **Active** (if a user is deactivated) or **Inactive** (to deactivate the user; see [Deactivating a User](#)).
5. Click **Save**.

Deactivating a user

Inactive users are prevented from logging in to your Sumo Logic account. You can reactivate a user at any time.

To edit a user:

1. Click the **Users** tab in the Sumo Logic Web Application.
2. If the **Manage Roles** page is displayed, click **View Users**.
3. Click **Edit** to the right of the user's name.
4. Select **Inactive** under Status.
5. Click **Save**.

Deleting a user

Deleting a user permanently removes the user's account, including his or her credentials. If you just need to temporarily prevent access, you can deactivate the user rather than deleting the account.

When you delete a user, roles assigned to the user are not affected, even if the roles aren't assigned to any other users.



Use caution when deleting a user. This action cannot be undone.

To delete a user:

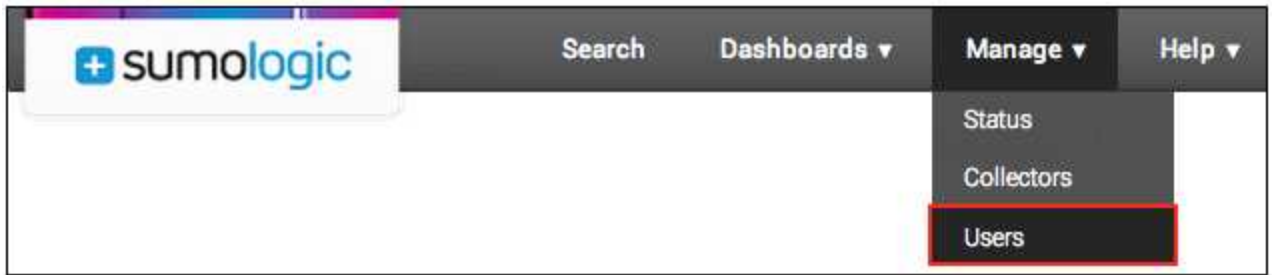
1. Click the **Users** tab in the Sumo Logic Web Application.
2. If the **Manage Roles** page is displayed, click **View Users** (to display the Manage Users page):
3. Click **Delete** to the right of the user's name.
4. Click **OK** to confirm that you want to delete this user.

Deactivating a user

To temporarily prevent a user from logging into the Sumo Logic service, you can change the user's status to Inactive. You can reactivate an inactive user at any time without the need to re-enter user details. If you wish to permanently remove a user, you can delete them.

To deactivate a user:

1. Click **Manage > Users** in the Sumo Logic Web Application.



2. Make sure the **Manage Users** page is displayed.
3. Click the **Edit** link next to the user's name to edit details for the user.
4. For the Status option, select **Inactive**.

5. Click **Save**. The user is not notified of the change.

To reactivate a user:

- Change the user's Status option back to **Active** and click **Save**.

Deleting a user

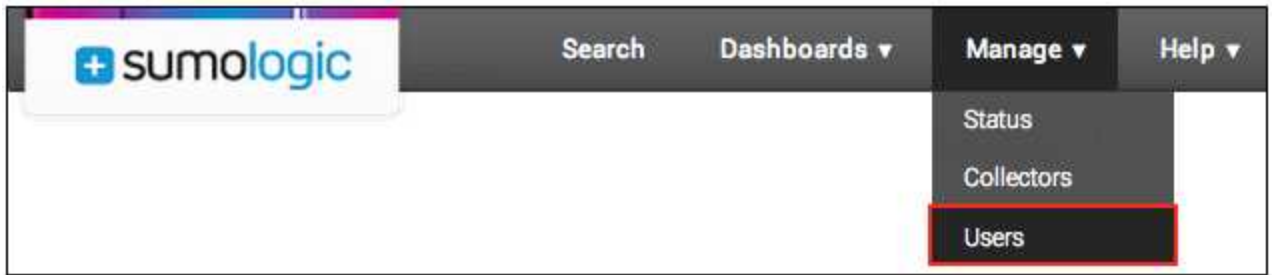
Deleting a user permanently removes the user and his or her associated settings, including scheduled and shared searches and shared Dashboards. To save the settings for a user and temporarily prevent access, you can deactivate a user rather than deleting them.



Use caution when deleting a user. This action cannot be undone.

To delete a user:

1. Click **Manage > Users** in the Sumo Logic Web Application.



2. Click the **Delete** link to the right of the user's name.

Manage Users				View Roles	New User	↻
Name	Email		Roles			
Alex	alex@demo.com	✓	Administrator	Edit	Delete	
Amanda	amanda@demo.com	✓	Administrator, Collector Management	Edit	Delete	
Ana	ana@demo.com	✓	Analyst	Edit	Delete	
Bill	bill@demo.com	✓	Administrator	Edit	Delete	

3. Click **OK** to confirm that you want to delete the user.

Resetting a user's password

The following procedure is for Admins to use to reset another user's password.



Looking for instructions on resetting your own password? See [Changing Your Password](#).

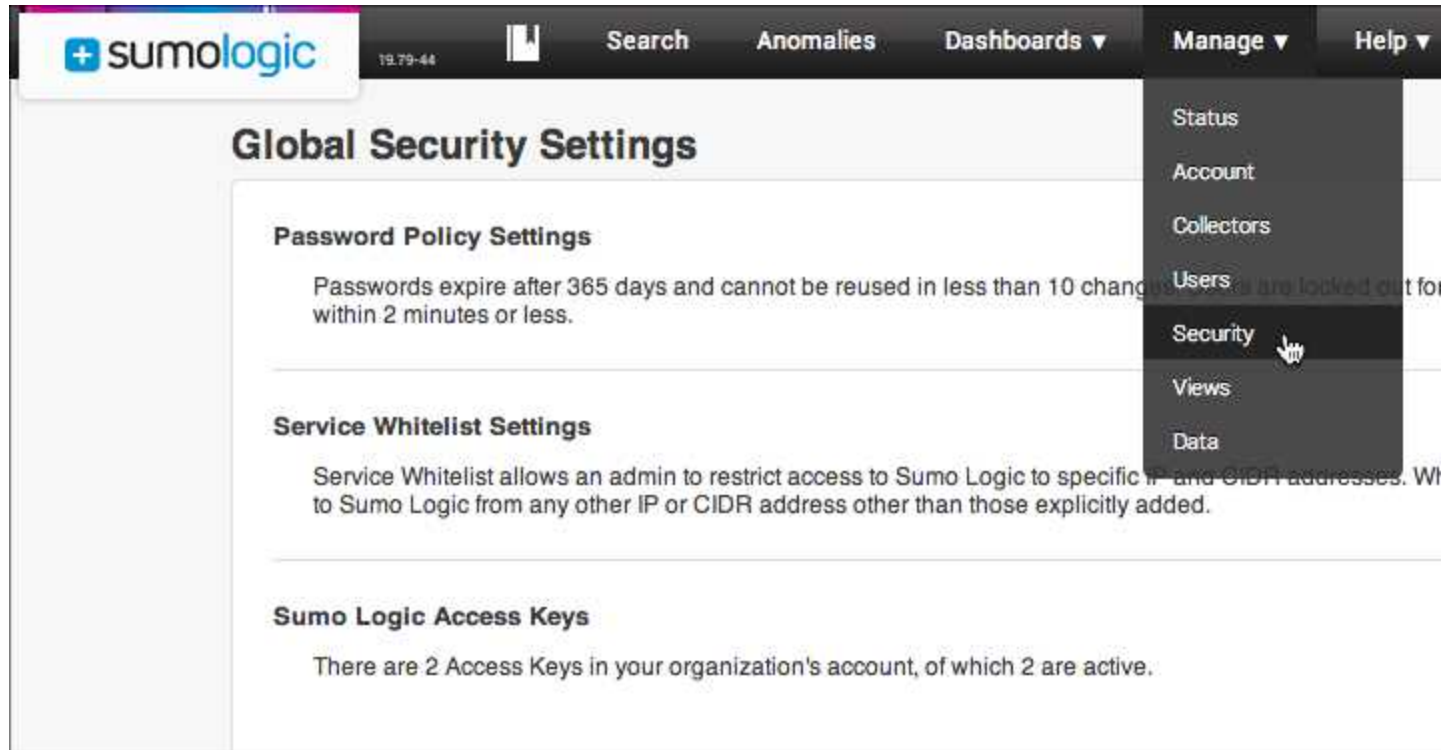
To reset a user's password:

1. Click the **Users** tab in the Sumo Logic Web Application.
2. If the **Manage Roles** pane is displayed, click **View Users**.
3. Click **Edit** to the right of the user's name.
4. At the bottom of the Edit User dialog, click the link for **Reset Password**.
5. Confirm the password reset by clicking **OK**.

A new random password is generated and sent to the user at the email address listed in the user's details. After the user logs in with the reset password, he is immediately prompted to enter a password of his own choosing.

Managing Security Settings

The **Security** page of the Sumo Logic Web Application makes it easy to manage security policies in one place. To access the Security page, go to **Manage > Security**.



From here you can manage:

- [Password settings](#).
- [IP and CIDR whitelisting](#).
- [Access Keys](#)

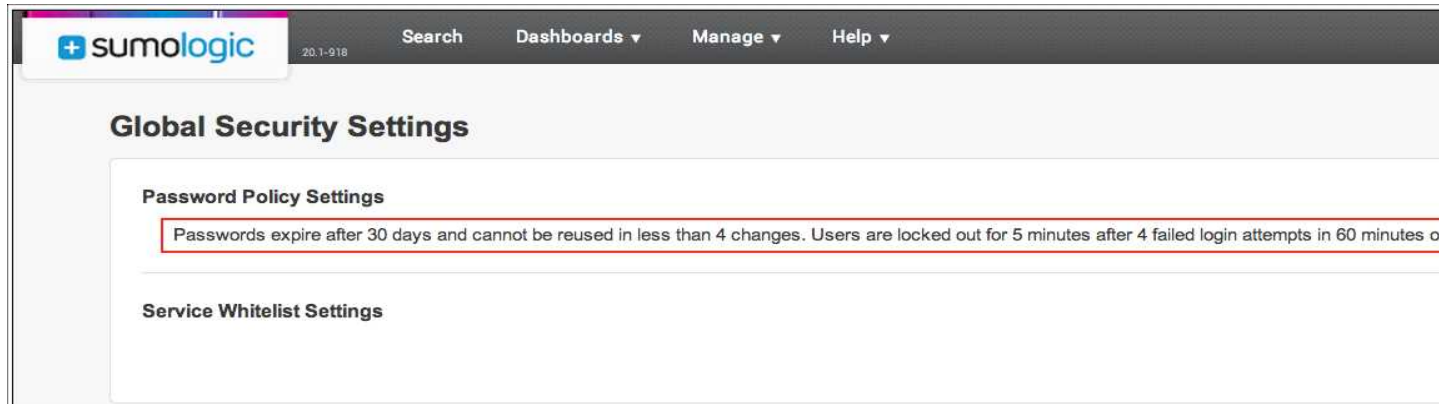
Setting password policies

Account Admins can set the account's password policy through the **Security** page. Several components can be customized:

- **Passwords expire in** sets the number of days before a password expires and must be reset. The minimum is 30 days; the maximum is 365 days.
- **Passwords reuse after** sets when a password previously used by a person can be used again.
- **Users locked out after** allows an admin to set when a user is locked out of his or her Sumo Logic account. The admin can choose the number of failed attempts, the amount of time during which the incorrect password is entered, and the amount of time a user will be locked out of his or her account after entering the set number of incorrect passwords.

Where can I view my organization's current Sumo Logic password policy?

The existing password policy can be viewed on the Security page:

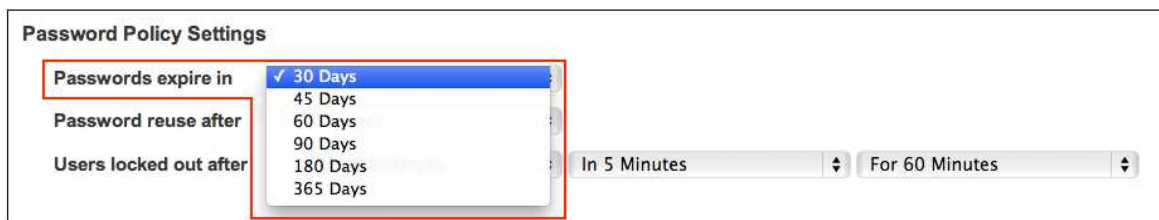


Changing password settings

Admins can make changes at any time for users in their organization. The updated settings are applied to each user's account the next time he or she logs in.

To change password policy settings:

1. On the **Security** page, next to **Password Policy Settings**, click **Edit**.
2. Change any of the following:
 - **Number of days before a password must be reset.** For **Passwords expire in**, choose an option from the menu.



- **Number of times a password must be changed before a previously used password can be reused.** For **Password reuse after**, choose the number of changes. For example, if you choose **5 Changes**, a password can be reused after five new passwords have been used in a user's account.

Password Policy Settings

Passwords expire in: 30 Days

Password reuse after: 4 Changes (selected), 5 Changes, 6 Changes, 7 Changes, 8 Changes, 9 Changes, 10 Changes

Users locked out after: In 5 Minutes, For 60 Minutes

- **User lockout settings.** With the **Users locked out after** options, you can determine when users are locked out of their Sumo Logic accounts using the three menus: number of failed attempts, amount of time during which the incorrect password is entered, and the amount of time a user will be locked out of his or her account after entering the set number of incorrect passwords.

For example, we chose **4 Failed Attempts** from the first menu, **Within 5 Minutes** from the second menu, and **For 60 Minutes** from the third menu. This means that if a user enters four incorrect passwords in the space of five minutes, that user will be unable to log back in to his or her account for 60 minutes.

Password Policy Settings

Passwords expire in: 30 Days

Password reuse after: 4 Changes (selected), 3 Failed Attempts, 5 Failed Attempts, 6 Failed Attempts, 7 Failed Attempts, 8 Failed Attempts, 9 Failed Attempts, 10 Failed Attempts

Users locked out after: Within 1 Minutes, Within 2 Minutes, Within 3 Minutes, Within 4 Minutes, Within 5 Minutes (selected), Within 8 Minutes, Within 10 Minutes

For 30 Minutes, For 45 Minutes, For 60 Minutes (selected), For 120 Minutes

4. Click **Save**.

Whitelisting IP or CIDR addresses

Service Whitelist Settings allow you to explicitly grant access to specific IP addresses and/or CIDR notations. By default this feature is not enabled. The IP address of the Admin who enables the feature is automatically added to the whitelist. Once the feature is enabled, the IP address or CIDR of each user in an account must be added to the whitelist in order to log in to Sumo Logic.

Users who are logged in when the whitelist is enabled will continue to be logged in; the settings take effect after a user has logged out of his or her Sumo Logic account. Any IP or CIDR addresses must be associated with your company in order to add them to the whitelist. Wildcards are not supported.

To enable Service Whitelist Settings:

1. On the **Security** page, to the right of **Service Whitelist Settings** click **Edit**.
2. Select **Enable Service Whitelist**, then copy and paste your IP address in the IP Address or CIDR text box and click **Add**.

Service Whitelist Settings

 Your current IP Address is 12.77.21.34, which isn't in the Service Whitelist.

Service Whitelist Status

☒ Enable Service Whitelist

If enabled, access to Sumo Logic is granted only to the IP addresses and CIDR ranges added to the whitelist.

IP Address or CIDR

12.77.21.34

Reset

Cancel

- Type additional IP and/or CIDR addresses in the text box, and click **Add**. Repeat this step until you've added all the addresses you'd like to whitelist.

Service Whitelist Settings

• 1.202.91.134	• 12.22.2.3	• 13.22.2.6	• 15.22.2.11
• 10.22.2.1	• 12.22.2.4	• 13.22.2.7	• 15.22.2.12
• 10.22.2.10	• 12.22.2.5	• 13.22.2.8	• 15.22.2.2
• 10.22.2.11	• 12.22.2.6	• 13.22.2.9	• 15.22.2.3
• 10.22.2.12	• 12.22.2.7	• 14.22.2.1	• 15.22.2.4
• 10.22.2.2	• 12.22.2.8	• 14.22.2.10	• 15.22.2.5
• 10.22.2.3	• 12.22.2.9	• 14.22.2.11	• 15.22.2.6
• 10.22.2.4	• 122.34.43.54	• 14.22.2.12	• 15.22.2.7

Service Whitelist Status

☒ Enable Service Whitelist

Checking this box enables the feature and may lock out other people if the list is not properly configured.

IP Address or CIDR

Add IP Address or CIDR

Reset

Cancel

- Click **Save**.

To disable Service Whitelist Settings:

- On the **Security** page, to the right of **Service Whitelist Settings** click **Edit**.
- Deselect **Enable Service Whitelist**.
- Click **Save**.

Editing or deleting whitelisted addresses

After an IP or CIDR address has been whitelisted you can edit or delete the address. Note that any edits are immediately put into effect; deletions are immediate and cannot be undone.

To edit a whitelisted address:

- Click an address, then make any edits in the text box.

2. Click **Edit**.



Service Whitelist Status

☒ Enable Service Whitelist

Checking this box enables the feature and may lock out other people if the list is not properly configured.

IP Address or CIDR

15.22.2.4

Reset Edit

Cancel Save

3. Click **Save**.

Changes are applied immediately.

To delete a whitelisted address:

1. Hover over the address, then click the "x".



Service Whitelist Settings

- 1.202.91.134
- 12.22.2.3
- 13.22.2.6
- 15.22.2.11 

2. Click **Save**.

Using Access Keys

Access Keys are generated by an individual user in the Sumo Logic Web Application; however, admins in the account can deactivate or delete all keys generated across the account. Access Keys are associated with the user who generated them; if a user is disabled, keys associated with that user will no longer work. **When a user is removed from the account their keys will no longer authenticate.**

Access Keys are used two ways:

To securely register new Collectors. When installing a Collector, you can choose to enter email and password credentials, or you can use the Sumo Logic Web Application to generate unique Access Keys. Access Keys can be generated and used per your organization's policies. You can generate just one set and use those credentials on all Collectors, or a new set can be generated for each Collector, depending on how your group prefers to manage this process. (Note that keys are only used upon installation, so if a key is deleted after a Collector has been set up, the Collector won't be affected.)

To access the service interface. Access Keys can be used with Sumo Logic's service APIs, including our Search API and Collector Management API.

How do I use Access Keys in Sumo Logic APIs?

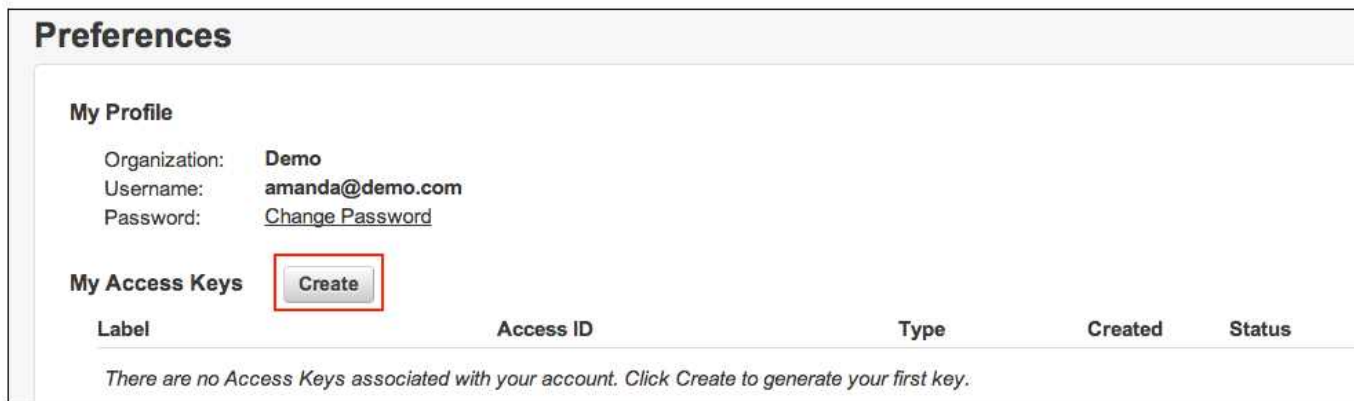
Access Keys can be used in place of email and password credentials in all Sumo Logic APIs. Instead of passing email and password parameters, you'll use the access key and access ID parameters. Please refer to the API documentation for more information. Links to all API documentation can be found [here](#).

Creating new Access Keys

New Access Keys can be generated at any time by any Admin in your organization's Sumo Logic account. Once keys have been added, they can also be copied and edited by any Admin.

To generate Access Keys:

1. In the Sumo Logic Web Application click your user name, then choose **Preferences**.
2. Next to **My Access Keys**, click **Create**.



Preferences

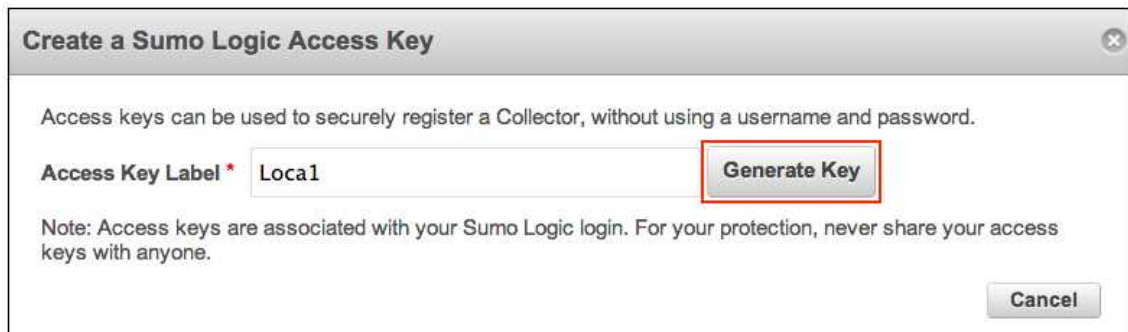
My Profile

Organization: **Demo**
Username: **amanda@demo.com**
Password: [Change Password](#)

My Access Keys **Create**

Label	Access ID	Type	Created	Status
There are no Access Keys associated with your account. Click Create to generate your first key.				

3. In the **Access Key Label** text box, type a name for the Access Key. Then click **Generate Key**.



Create a Sumo Logic Access Key

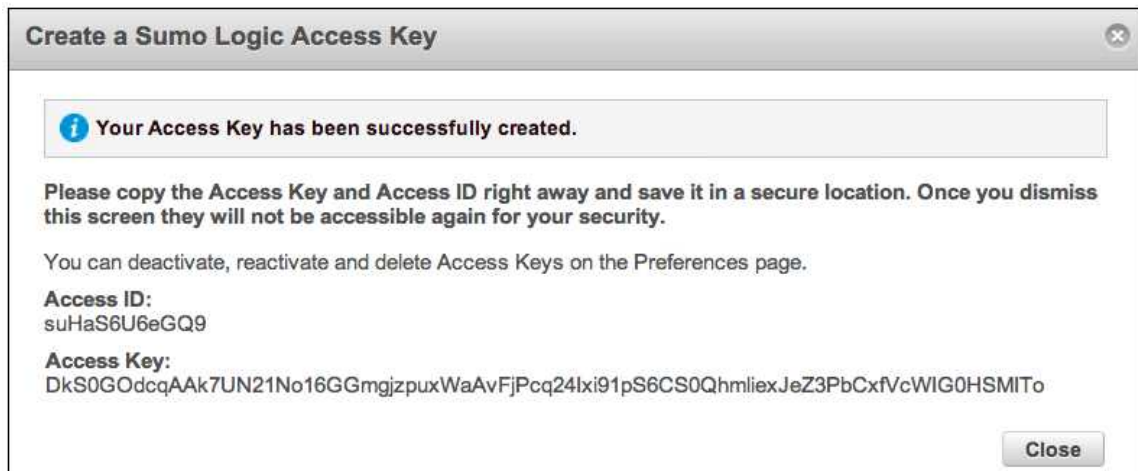
Access keys can be used to securely register a Collector, without using a username and password.

Access Key Label * Local **Generate Key**

Note: Access keys are associated with your Sumo Logic login. For your protection, never share your access keys with anyone.

Cancel

4. When the new key and ID is displayed, copy the credentials right away and paste them in a safe place. **Once you dismiss this screen you won't have full access to the key/ID again.**



Managing Access Keys

Keys that you generated can be found on your **Preferences** page. Admins in an account can deactivate, reactivate, or delete all keys in an organization through the **Security** page.

On the **Security** page, under **Sumo Logic Access Keys**, click **Edit**.

To deactivate an Access Key:

Deactivating a key keeps the key credentials in your account, but renders the key useless. You can reactivate a key at any time to begin using it again.

- Click **Active** in the Status column.

My Access Keys		Create				
Label	Access ID	Type	Created	Status		
Collector01	suQd6DizQZPNss	Collector	10/16/2013	Active	Delete	

To reactivate an Access Key:

- Click **Inactive** in the Status column.

My Access Keys		Create				
Label	Access ID	Type	Created	Status		
Collector01	suQd6DizQZPNss	Collector	10/16/2013	Inactive	Delete	

To delete an Access Key:

- Click **Delete** and then confirm the deletion. The key is immediately removed, and will no longer work.

My Access Keys						
		Access ID	Type	Created	Status	
loca1		suwweqxwrZUS77	Collector	11/01/2013	Active	Delete

Enabling a Support Account

Administrators can decide to enable a Sumo Logic support account, which grants very select Sumo Logic support agents access to your organization's account, better helping those agents to resolve issues that arise. Admins can choose to keep the Support Account enabled full-time, or the account can be disabled when no issues are being investigating.

When a support account is enabled, a special user is added to your organization's Sumo Logic account, named Sumo Logic Support. This is the user that Sumo Logic support agents will use to log in to your organization's account to troubleshoot issues. If you disable your support account, the Sumo Logic Support user account is disabled. It's important to remember to capture any content created by the Sumo Logic user account before disabling it.

About Support Accounts

Who can access my Support Account?

All requests to access an organization's support account are vetted by our Director of Security. When a support agent requests access, he or she is asked for the explicit reason why access is required. Additionally, the amount of time the agent can access the support account is limited to the shortest amount of time necessary to complete the investigation.

Do I need to create a special user account?

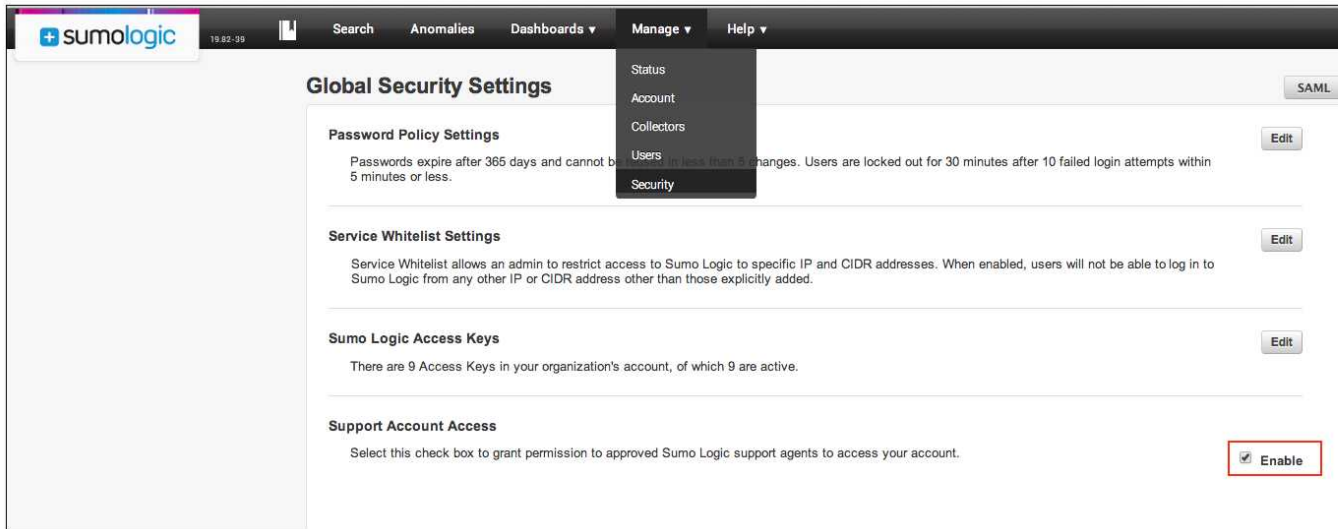
No; the Sumo Logic Support user account is automatically added to your organization's account. If you accidentally delete the user, you can just disable, then re-enable the support account and the Sumo Logic Support user will be recreated. However, any content created or shared from the previous iteration of the Sumo Logic Support user will be deleted.

Enabling your organization's Support Account

Support accounts are managed through the Security page of the Sumo Logic Web Application.

To enable a Support Account:

1. In the Sumo Logic Web Application, choose **Manage > Security**.
2. Click Enable for **Support Account Access**.



Click **Enable** again to disable the support account.

Managing Billing

The Billing page is available to the Account Owner of a **Sumo Logic Professional** or a **Sumo Logic Free** account. You can manage the following using the Billing page:

- Change the Account Owner role.
- Change billing information.

If your organization has a **Sumo Logic Enterprise** account, all billing is handled through your Sumo Logic Sales representative. If you're not sure who to contact, please drop us a line at sales@sumologic.com.

Upgrading your Account

If your organization currently has a **Sumo Logic Free** account, or if you've already upgraded to a **Sumo Logic Professional** plan you can upgrade to a more robust plan at any time. Sumo Logic Professional accounts allow up to 20GB of data volume. All Professional accounts include 20 users, 30 days of data retention, and access to [SAML](#) and [data forwarding](#).

Upgrades are processed immediately—meaning that your organization can take advantage of having a higher data volume right away.



Upgrades to Enterprise accounts are handled through a Sumo Logic sales representative. Not sure who to contact? Send an email to support@sumologic.com.

Who can upgrade my organization's account?

It depends on your current account type:

- If your organization currently has a **Sumo Logic Free** account, any Admin can perform the upgrade. That Admin is then known as the **Account Owner**, meaning that he or she is the only person in the account that can view and change the credit card information billed by Sumo Logic. For more information about Account Owner privileges, see [Using the Billing Page](#).
- If your organization already has a **Sumo Logic Professional** account, only the existing Account Owner can perform upgrades.
- **Sumo Logic Enterprise** accounts do not have Account Owners, as all upgrades are handled by a Sumo Logic salesperson.

Can I upgrade more than once?

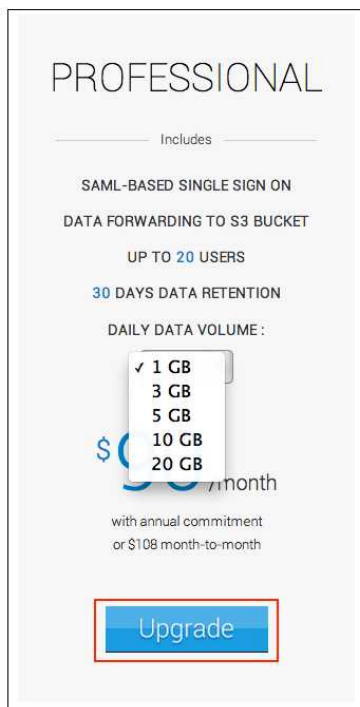
Yes. An Admin can upgrade from Sumo Logic Free to Sumo Logic Professional. Or, an Account Owner can upgrade a Sumo Logic Professional account to a higher level of data volume, up to 20GB. If you need additional data volume, you can simply request to be upgraded to a Sumo Logic Enterprise account by contacting sales@sumologic.com.

Performing an upgrade

Before beginning the upgrade make sure that you have an approved, valid credit card number. You'll need to provide payment information during the upgrade process.

To upgrade your account:

1. Do one of the following to access the Upgrade page:
 - In the **Welcome** screen, click **Upgrade Your Account**.
 - In the **Account** page, click **Upgrade Your Account**.
 - Click **Upgrade** at the top of the Web Application.
2. In the Upgrade page, choose a Daily Data Volume option (1GB, 3GB, 5GB, 10GB, or 20GB). The pricing information updates as you choose different options.
3. When you've chosen a Daily Data Volume option, click **Upgrade**.



4. In the **Payment Method** screen, enter the credit card information you'd like Sumo Logic to bill, or if you've previously upgraded you may choose to use the existing payment information. Click **Next**.
5. In the **Confirm Upgrade** screen, choose how often you'd like to be billed for your account, either annually (one lump sum) or monthly(12 equal payments). Review all the information displayed; if you'd like to change the plan you've selected, click **Select a Plan** at the top of the page.

1SELECT A PLAN

2SELECT A PAYMENT METHOD

3CONFIRM UPGRADE

CONFIRM UPGRADE

Your payment options

☒ Annually - \$1080☐ Month to month - \$108

Order Summary

You are currently upgrading your Sumo Logic account to "Professional" with the following changes. By applying these changes your contract will be automatically renewed with today's date.

- 1 GB Data Volume
- 30 Days Retention

Total: **\$90/mo (billed annually)***

Click "Confirm" to confirm that you would like to upgrade your account.
A payment of \$1080 will be processed and your account will automatically renew every year.

*Data Overage fees will be applied when you exceed your current plan. When you send more data than your current subscription, Sumo Logic will continue to receive and process data at four times your current per GB price.

Confirm

NEW PLAN

[EDIT](#)

PROFESSIONAL

- 1 GB Data Volume
- 30 Days Retention

PAYMENT

[EDIT](#)

Total: \$90/mo (billed annually)*
Visa ending with 3456

6. Click **Confirm** to complete the upgrade. After you click Confirm the credit card you provided to Sumo Logic is charged.
7. The upgrade is processed, then a Congratulations screen appears. Click **Finish**.

If you have any issues, or if you don't see a charge on your credit card within 48 hours, please contact support@sumologic.com.

Using the Billing Page

The Billing page is a tool made available only to the Account Owner of a **Sumo Logic Professional** account. Account Owner is solely responsible for the billing information provided to Sumo Logic. The Account Owner can reassign this role to any other Admin in his or her organization.

Do other account types have Account Owners?

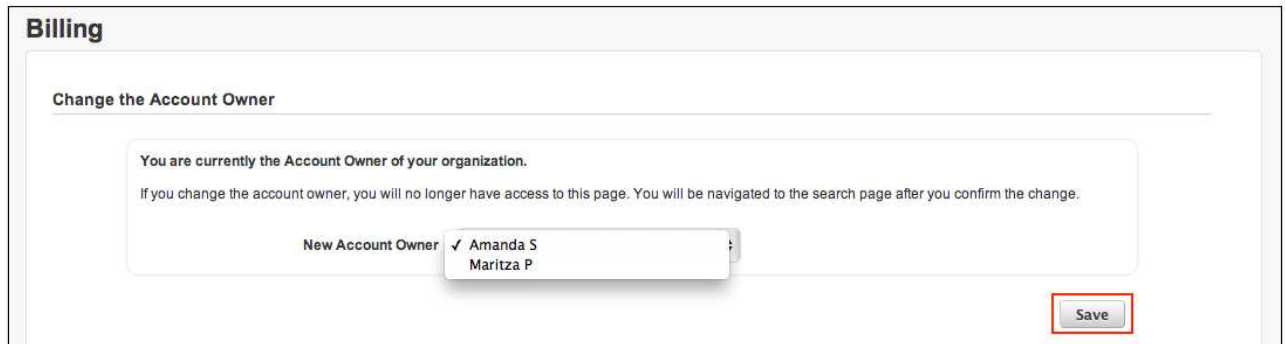
Yes, Sumo Logic Free accounts also have an Account Owner. By default the Account Owner in a Sumo Logic Free is the Admin who set up the account/signed up for the account. An organization with a **Sumo Logic Enterprise** account does not have an Account Owner, as all upgrades are handled by a Sumo Logic salesperson; no billing information can be changed via the Sumo Logic Web Application.

Changing the Account Owner

The Account Owner can reassign the role to any other Admin in your organization's Sumo Logic account.

To change the Account Owner role:

1. Click your user name, then choose **Billing**.
2. Choose a name from the **New Account Owner** menu, then click **Save**.



The screenshot shows a web interface titled "Billing". Below the title is a section "Change the Account Owner". Inside this section, there is a message: "You are currently the Account Owner of your organization. If you change the account owner, you will no longer have access to this page. You will be navigated to the search page after you confirm the change." Below the message is a dropdown menu labeled "New Account Owner" with two options: "Amanda S" (selected) and "Maritza P". A "Save" button is located at the bottom right of the section.

3. After the Account Owner has been changed, only the new Owner will have access to the Billing page.

Managing Billing information for Professional Accounts

Only the Account Owner can manage the billing information used by Sumo Logic. This includes the credit card number on file (used to pay monthly or annually) as well as the billing address/contact information.

Once changes are submitted Sumo Logic will begin using the new credit card for the next billing cycle.

To change the billing information:

1. Click your user name, then choose **Billing**.
2. Click Use a New Credit Card.
3. Enter the new credit card information, or edit any of the billing address/contact information.





4. Click **Submit**.

Billing

☐ Use Existing

Visa *****3456 , expires 1/2020

☒ Use a new Credit Card (This will replace the credit card on file)

CARD TYPE ☒ VISA    

CARD NUMBER 4111111111111111

EXPIRATION DATE 01 / 2017

CVV 111

CARDHOLDER NAME Amanda S

COUNTRY United States

ADDRESS 1 305 Main Street

ADDRESS 2

CITY Redwood City

STATE CA

POSTAL CODE 94063

CONTACT PHONE NUMBER 6508108700

EMAIL ADDRESS amanda@sumo.com

Submit

Managing Ingestion

The rate of data creation is rarely constant. Whether your organization sees seasonal spikes, or if a new feature or product line produces huge increases in activity, Sumo Logic meets the needs of your organization, known or unknown, while maintaining the search performance you rely on.

When designing your deployment, it's important to consider how logs will be ingested across Collectors in your account.

Throttling

Part of managing spikes in activity is properly slowing the rate of ingestion while the demand is at its peak, known as **throttling**.

Throttling is enabled across all Collectors in an account. Sumo Logic measures the amount of data already committed to uploading against the number of previous requests and available resources (quota) in an account. Although the rate of uploads may be slowed, logs remaining on a Collector will be uploaded over an appropriate amount of time.

Throttling also keeps one Collector from uploading more data than others—to the point where all data is being ingested from one Collector.

Account caps

Additionally, Sumo Logic imposes account caps on uploads to better protect against huge, unexpected overages in an account. By default, organizations are allowed to exceed between 2 times to 5 times the daily maximum (depending on account size). In general terms, this means that a paid 500GB account can spike up to 1.5TB. Again, even if the cap is exceeded, log data is kept safely at the Collector level until quota is made available, at which time the data is ingested.

Sumo Logic free accounts can expect slight different behavior. If a Sumo Logic Free account regularly exceeds the cap, the account is temporarily disabled until quota becomes available (or until the account is upgraded).

Sumo Logic accounts can be upgraded at any time to allow for additional quota.

Provisioning SAML

Sumo Logic now supports self-provisioning of Security Assertion Markup Language (SAML) to enable Single Sign-On (SSO). In addition to basic SAML configuration, you can choose optional on-demand user creation (via SAML 2.0 assertions), and designate custom log in and/or log out portals.

SAML is supported for Sumo Logic Enterprise accounts. If your organization has a Sumo Logic Free account, please [contact us](#) to upgrade.

How does logging in work after provisioning SAML?

After SAML is up and running, users in your organization will have a few options for logging in. First, users can log in through your IdP, just as they log in to other URLs now. Alternately, users will still be able to log in to the Web Application using their Sumo Logic credentials. Sumo Logic still provides each user a password even for organizations that use On-Demand Provisioning. Or, if you configure an SP Initiated Login path, users can log in through that designated URL with a single click.

Prerequisites

Before provisioning SAML, make sure you have the following:

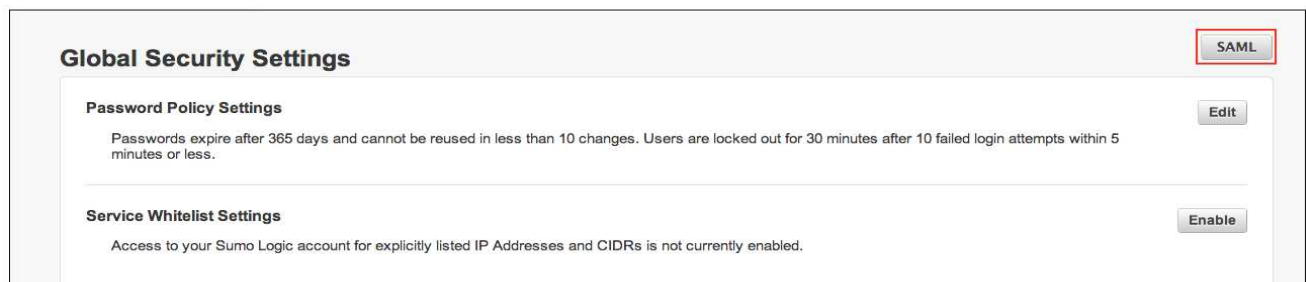
- **Identity Provider (IdP).** There are several SAML IdPs available. As long as your organization's IdP supports SAML 2.0 you can configure SAML in Sumo Logic.
- **X.509 certificate.** This certificate is used to verify the signature in SAML assertions. Up to three certificates can be specified.

Configuring SAML

SAML configuration is handled through the Account tab of the Sumo Logic Web Application. Most of the required information can be gathered from your IdP; other options should be covered by your internal access policy. If you need to edit any settings in the future, you'll use the same dialog box.

To configure SAML:

1. In the Web Application, choose **Manage > Security**.
2. In the Security page, click **SAML**.



3. Click **Configure**.
4. Enter the following:

Configure SAML 2.0

Configuration Name *

?

☒ Debug Mode

Issuer *

Authn Request URL

X.509 Certificate *

Email Attribute

☐ Use SAML subject

☒ Use SAML attribute

Roles

☒ SP Initiated Login Configuration

Login Path

☐ On Demand provisioning (Optional)

☒ Logout Page (Optional)

URL

https://www.google.com

☒ Roles Attribute (Optional)

Cancel

Save

Configuration Name. Type the name of the SSO policy (or another name used internally to describe the policy).

Debug Mode. Select this option if you'd like to view additional details when an error occurs. For more information, see [Using SAML Debug Mode](#).

Issuer. Type the unique URL associated with your organization's SAML IdP.

Authn Request URL. Type the URL Sumo Logic will use to submit SAML authentication requests to your IdP.

X.509 Certificate. Copy and paste your organization's X.509 certificate, which is used to verify signatures in SAML assertions.

Email Attribute. Depending on your IdP, select **Use SAML subject**, or select **Use SAML attribute** and then type the email attribute name in the text box.

SP Initiated Login Configuration. Type a unique alphanumeric identifier string to generate an SP Initiated Login path. For example, typing SamlDemo in the text box would generate <https://service.sumologic.com/sumo/saml/get/SamlDemo> as the SP Initiated Login.

On Demand Provisioning. Select this option to have Sumo Logic automatically create accounts when a user first logs on. For more information, see [Optional SAML features](#).

Logout Page. Select this option if you'd like to point all users to a specific URL after logging out of Sumo Logic. For more information, see [Optional SAML features](#).

Roles Attribute. To have roles assigned the first time a new user logs in for the first time, select **Roles Attribute**, then type the role names. These role names must match the roles established in your IdP. For more information, see [Optional SAML features](#).

5. Click **Save**.
6. After the configuration settings are saved, the following information is displayed. You'll provide one of the SP Initiated URLs to your IdP:
 - If your IdP requires **HHP-POST binding**, copy the **POST URL** and copy it in your IdP's site.
 - If your IdP requires **HTTP-REDIRECT binding**, copy the **Redirect URL**, and then copy it in your IdP's site.

Configure SAML 2.0

Select a configuration or create a new one ⓘ

Test SAML ▼

SP Initiated
 Redirect: `https://service.sumologic.com/sumo/saml/get/myName`
 POST: `https://service.sumologic.com/sumo/saml/post/myName`

Authentication Request
`https://service.sumologic.com/sumo/saml/login/1102333023`

Assertion Consumer
`https://service.sumologic.com/sumo/saml/consume/1102333023`

Delete Configure

Can I have more than one SAML configuration?
 You can create a configuration for each SSO implementation your organization uses. If you have a single SSO implementation, you don't need more than one SAML configuration. [Learn more...](#)

The Authentical Request URL is the Authn Request URL you specified during configuration.

7. Click the X to close the dialog box, or if you need to make any edits, click the Configure button.

Optional SAML features

When configuring SAML, there are two optional features that you can use to set up how new user accounts are added, and where users are directed after logging out of Sumo Logic.

Setting up On Demand provisioning

In most cases, setting up On-Demand provisioning makes adding new users very simple: the first time a user signs in to Sumo Logic, an account is automatically created.

Prerequisites:

Specify Sumo Logic roles. In order for new accounts to be created, you'll need to specify Sumo Logic RBAC roles in the [SAML Configuration dialog box](#).

SAML Attributes. You'll need the First Name and Last Name attributes your IdP uses to identify users.

When the account is created, Sumo Logic credentials are emailed to the user. (Users need both Sumo Logic credentials and SAML permissions.)

☒ **On Demand provisioning (Optional)**

First Name Attribute

firstName

Last Name Attribute

lastName

On Demand Provisioning Roles*

Analyst

Adding a Logout Page

Adding a **Logout Page** sets a specific URL where users in your organization are sent after logging out of Sumo Logic. You could choose your company's intranet, for example, or any other site that you'd prefer users in your organization access.

To add a log out page, type the URL in the dialog box:

☒ **Logout Page (Optional)**

URL

https://www.google.com/

Using roles set up via your IdP

When enabled, the **Roles Attribute** option assigns roles to a new user the first time the user logs in. These roles are configured via your IdP. Because the assertions are provided to Sumo Logic by your IdP, the role names you enter in the Roles Attribute text box must exactly match those names in the IdP.

Viewing SAML debug information

When the Debug Mode option is selected, whenever an error occurs a debug page is displayed. The debug page is displayed in a new browser window (not within the Sumo Logic Web Application). A new page is generated for each error.



SAML 2.0 Debug Information

Errors

User SamIDebugPage@demo.com not found!

Parsed response data

```
request
success: true
in response to: null
assertion
issuer: https://app.onelogin.com/saml/metadata/82568
one time use: false
conditions: 1,
times
validity: 2013-02-19T09:57:18.000-08:00 - 2013-02-19T10:03:18.000-08:00
valid now: true
```

Response XML

```
<?xml version="1.0" encoding="UTF-8"?> <samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" Destination="[recipient]"
ID="GOSAMLR13612968187173" IssueInstant="2013-02-19T18:00:18Z" Version="2.0" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
<saml:Issuer>https://app.onelogin.com/saml/metadata/82568</saml:Issuer>
<samlp:Status>
<samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</samlp:Status>
<saml:Assertion ID="pfxd53cb50b-b4ea-b717-f3da-7506a36cb158" IssueInstant="2013-02-19T18:00:18Z" Version="2.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<saml:Issuer>https://app.onelogin.com/saml/metadata/82568</saml:Issuer>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#pfxd53cb50b-b4ea-b717-f3da-7506a36cb158">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>SnaITHegW/LAxs7glxue6RIPzaA=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
```

Response XML (raw)

```
<samlp:Response xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" ID="GOSAMLR13612968187173"
Version="2.0" IssueInstant="2013-02-19T18:00:18Z" Destination="[recipient]">
<saml:Issuer>https://app.onelogin.com/saml/metadata/82568</saml:Issuer> <samlp:Status> <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/> </samlp:Status> <saml:Assertion xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Version="2.0" ID="pfxd53cb50b-b4ea-b717-f3da-7506a36cb158" IssueInstant="2013-02-
19T18:00:18Z"> <saml:Issuer>https://app.onelogin.com/saml/metadata/82568</saml:Issuer> <ds:Signature
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"> <ds:SignedInfo> <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/> <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/> <ds:Reference URI="#pfxd53cb50b-b4ea-b717-f3da-
7506a36cb158"> <ds:Transforms> <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/> <ds:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/> </ds:Transforms> <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/> <ds:DigestValue>SnaITHegW/LAxs7glxue6RIPzaA=</ds:DigestValue> </ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>FdDP7I89mi5dEzVyxjqezG3ITWsK0xtCpeiTyOI89ZQj1kZeOfmk9YZs1Fb8P9ska4XX7ADG8JG7nJD/Ty8yMk5JWyuQ6h/FuSO1DihxhpZL8lw6r
3kZPupMyLVjvwwZTGHVOKvfhgBbAwJLHu/kor/LBNKmQcO1LLrgaOZgh4y24zJbn+V/rZLO+CmhETbXwNuGfCkqufltdAdpfxZ+zrmNEBVA8IG8HvBEaldjbXvmxE3
Hjph7ghCh7UjPDmYjCzJoZh2CYGXRZ/TJlVoz0QpoiY07Kj/E40rulqz2iuwXNVuUTsaDQo9aHWD8iDMFIW2hutXPQFhrZ7cg3Q==</ds:SignatureValue>
<ds:KeyInfo> <ds:X509Data>
```

The Debug page includes:

- **Error.** The actual error that triggered the debug page.
- **Parsed response data.** Subset of data that summarize the error condition.
- **Response XML.** Output of the error format in a human-readable form.
- **Response XML (raw).** Raw XML output generated by the error.

About Anomaly Detection

Anomaly Detection uses machine learning and logic to detect abnormalities in your environment while examining logs as they are ingested into Sumo Logic.

Anomaly Detection first uses LogReduce to assign logs to **Signatures**. Think of Signatures as sets of logs that are grouped together by commonality—not all logs in a Signature may match, but they are similar enough to logically be grouped together. Anomaly Detection then watches the general distribution of Signatures as your logs are ingested over time. Once Signatures appear in the Anomalies page, Admins can teach Sumo Logic how to handle Events by tagging with varying degrees of importance, or as Unimportant—removing the “noise” that can make finding unexpected activity so difficult.

Once Anomaly Detection has sufficient knowledge about the baseline behavior of your logs, abnormal deviations from the baseline are detected, then displayed in the Anomalies page of the Sumo Logic Web Application as Events, which is an indicator that Anomaly Detection has noticed activity that warrants additional attention.

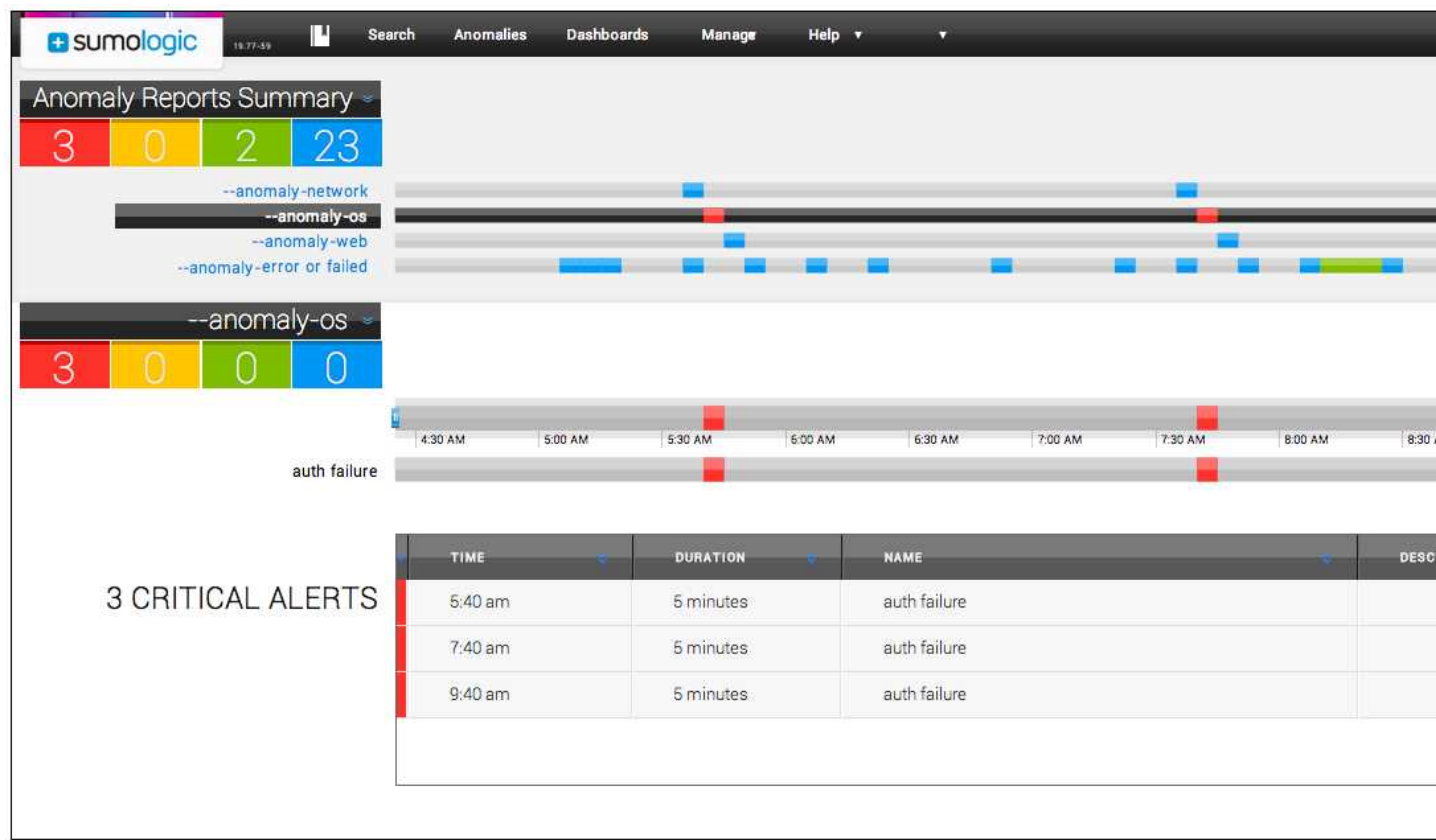
Who can use Anomaly Detection?

Anomaly Detection is available only to organizations with Enterprise accounts. If you currently have a Sumo Logic Free or Sumo Logic Professional account you'll need to upgrade in order to use Anomaly Detection.

About the Anomalies Page

The **Anomalies** page graphically breaks down the last six hours of activity in your Sumo Logic account using Reports that you'll define.

At the top of the page you'll find the **Summary Report**, which shows a consolidated view of all the individual Anomaly Reports. (Until you define one or more Reports, the Summary Report won't contain much activity.) Below the Summary you'll see the selected individual report, which displays the Incidents pertaining to one particular Event in your deployment.



Adding Reports to the Anomalies Page

Reports are defined by the set of logs that you tell Anomaly Detection to watch by specifying a filter-style query. Queries that produce aggregate results cannot be used in an Anomaly Report. For example, the following queries can be saved as a Report:

- `_sourceCategory=frontend`
- `_sourceCategory=frontend and _sourceHost=frontend-5`
- `_sourceCategory=frontend | parse "module=*" as module | where module="service"`

You can add up to 15 Anomaly Reports. After you create a new report, it takes some time for Anomaly Detection to learn the baseline behavior for that report—generally around six hours. During this time, no anomalies can be detected.



Certain queries can't be used to define Reports, including queries that generate aggregate results. Additionally, queries that include a summarize operator can't be used to create a Report.

To define a Report:

1. On the **Anomalies** page, click the double-arrow to the right of the **Anomaly Reports Summary**, and click **New**.



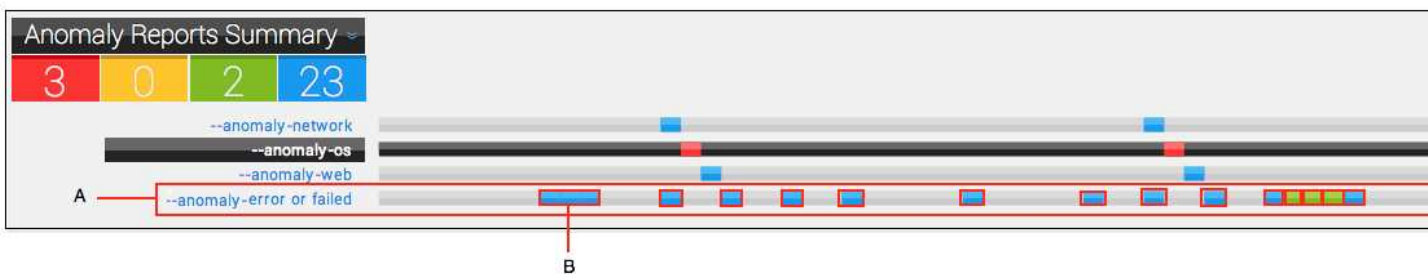
2. In the **New Report** dialog box, type a name for the Report, and then type the query you'd like to save as a report. Click **Save**.

About Events and Incidents

The colored blocks in a report are referred to as **Events**. An Event is associated with the query saved as an Anomaly Report. When an anomaly is detected it's saved as an **Incident**. Think of an Incident as an individual occurrence of an Event.

Sumo Logic captures Incident data, allowing you to easily drill down to a very granular level, giving you the exact time frame of an Incident, so your team can immediately react to any situation.

After an Event has been added to an Anomaly Report, you can drill down into each Incident to understand what triggered it, then tag the severity of the Incident or Event to help Anomaly Detection learn which behaviors are expected, which behaviors are unimportant, and behaviors that mean trouble.



A. Event.

B. Incident. Each Incident is one occurrence of the Event associated with the Anomaly Report.

Why are Events shown in different colors?

Anomaly Detection uses four different colors so you can tell each Event type apart. You can see the number of each event type in the Summary Report:



Color	Description
Blue	Unknown/unranked Event. The first time Anomaly Detection encounters an Event, it's tagged with blue. Seeing a blue Event is a signal that the Event needs to be investigated, named, and ranked.
Red	High severity.
Yellow	Medium severity.
Green	Low severity.



Events that you've tagged as Unimportant are initially displayed in gray, but are not shown in future reports.

Naming and Labeling Events

After investigating a new Event, you can name and rank the Event so Anomaly Detection learns how to handle this Event in the future.



Event names and ranks are applied across your organization's Sumo Logic account, not just your own login.

To name and rank an Event:

1. Click an **Event** in a Report. Remember that blue Events have not been named or ranked. Events that are red, yellow, or green have been labeled previously, but the rank can be changed.
2. After investigating the Event, type an **Event Name** for the Event (we used Intrusion), then choose a Severity setting.

sumologic 19.03-17 Search Anomalies (beta) Dashboards Manage Help

Frontend - Service
Intrusion

Event Name: Intrusion Description: Type description of this event

Severity: High (dropdown menu open showing: High, Medium, Low, Unimportant)

Signatures

#	Score	
1	480 +1,729%	\$DATE INFO [hostId=kwan-frontend-1] [module=SERVICE] [localUserName=service] [logger=service.util.interceptor.LoggingInterceptor] [thread=qtp90] Invocation: 'ReflectiveMethodInvocation: public abstract com.sumologic.service.endpoint.auth.v1.model.LoginResult'

Reset Save

Choosing Unimportant for the Severity means that you won't be notified if this Event appears again; it's treated like background noise.

3. Type an optional **Description** for the Event. This can include directions on how to handle the Event in the future. Including a Description can be helpful to others in your organization.

The Description field can be used as a transcript of how the issue was addressed/information about handling the issue in the future. Or, you can include contact information for the person who should handle escalations, for example.

4. Click **Save**.

Drilling Down into Events

There are two ways to gain more insight into an Event. To see a high-level view of the Signatures related to the discrepancy, click an Event. Or, for more granular information, you can view the log messages assigned to the Signature.

On the Event page, you'll also be able to take a look at Score and Change values for the Event. The size of the blue dot in the Score column directly relates to the relevancy of the Signature to the Event. (The more the Signature matches the Event, the larger the dot will be.) The Change values indicate if there are new matching Signatures, no matching Signatures, or if a Signature is more or less important to the Event.

To drill down into an Event:

1. Click the Event in a Report.
2. Do any of the following:
 - Review the Signatures assigned to the Event to discover what triggered the Event.
 - To review additional information, click the triangle to the left of the Signature to view the log messages associated with the Signature.
 - Check the Score and Change columns to check to see if the Event is new, if it's increasing or decreasing, or if there's no change.

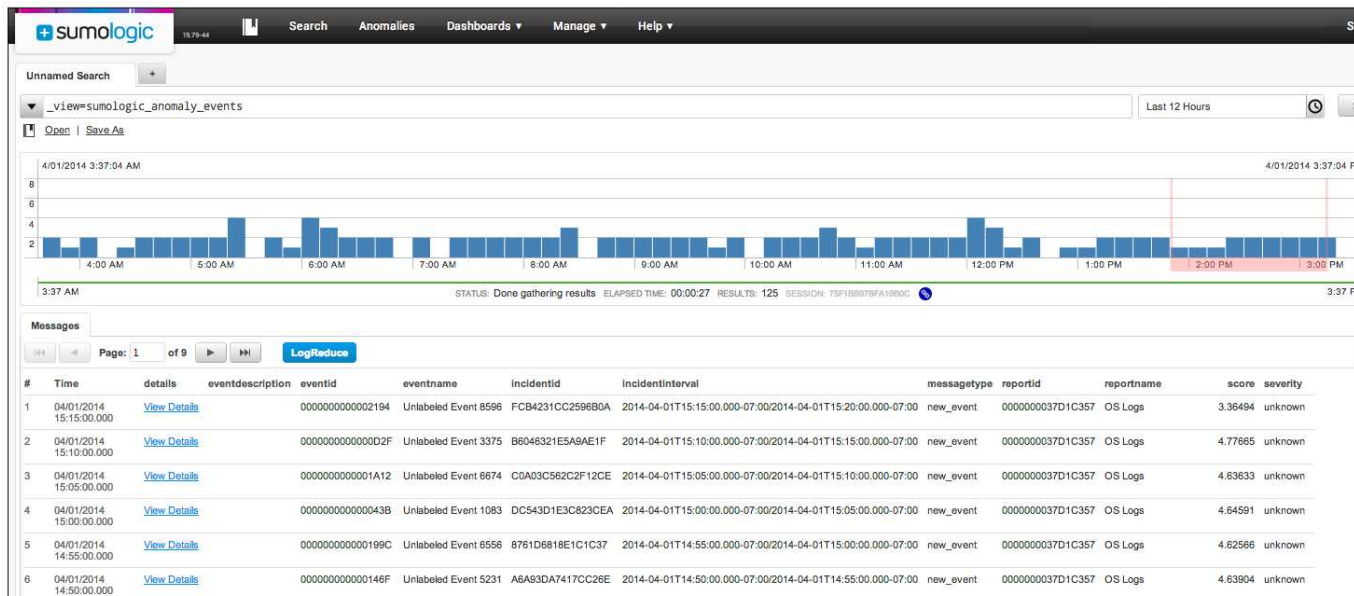
Viewing Historial Incidents

The Anomalies page displays incidents occurring over the most recent six hours. What if you'd like to review incidents that happened 12 or 24 hours ago, or even a month ago? Sumo Logic retains data associated with incidents that you can review by running a query on-demand, or even as a scheduled search.

This data is indexed in such a manner that historical search results are returned very quickly. Because of the indexing, you'll have to run a specific query to view historical Incidents.

To view historical Incidents:

1. In the Search page, enter the following query: `_view=sumologic_anomaly_events`
Make sure to enter the query exactly as shown. Changing any part of the query renders it ineffective.
2. Choose the time range for the Incidents that you'd like to review. Then click **Start** to run the search. The results display, similar to:



3. To view additional information about an incident, click **View Details**, which launches the Signatures pane for the Incident.

sumologic

10.79.44

Search

Anomalies

Dashboards ▾

Manage ▾

Help ▾

StockTrader - Web App

←

Successful Login after Multiple Failed Logins

Event Name

Successful Login after Multiple Failed Logins

Time Range

04-01-2014 16:00:00 to 04-01-2014 16:05:00

Severity

High

Description

Successful Login after Multiple Failed Logins

Signatures

#	Score	Change	Signature
1		↑	Mon \$DATE UTC 2014 username=matthew,login=failed
2		↓	Fri \$DATE UTC 2014 192.33.31.* GET /Trade/StockTrade.aspx action=*&symbol=s:* 80 bender *.*.* Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_3)+AppleWebKit/536.5+(KHTML,+like+Gecko)+Chrome/19.0.1084.54+Safari/536.5 200 0 0
3	*	↓	Tue \$DATE UTC 2014 192.33.31.131 GET /Trade/StockTrade.aspx action=sell&symbol=s:433 80 james 65.98.119.36 Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_3)+AppleWebKit/536.5+(KHTML,+like+Gecko)+Chrome/19.0.1084.54+Safari/536.5 200 0 0
4	-	↑	Tue \$DATE UTC 2014 username=matthew,login=failed
5		⌛	Sun \$DATE UTC 2014 username=peter,login=success
6		↑	Sat \$DATE UTC 2014 username=kumar,login=success

You can then investigate the signatures associated with the Incident as you'd like.

Assigning Anomaly Detection Permissions to Users

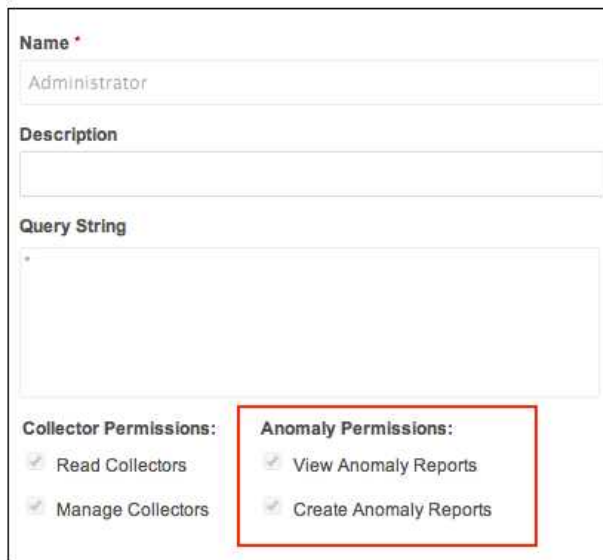
To ensure that only specific users can view and/or create Anomaly Reports, those permissions can be added to or disabled from roles in your Sumo Logic account. The two Anomaly Permissions are:

View Anomaly Reports. Allows a user to view the Anomaly Reports saved in your organization. Users with this role are unable to edit Anomaly Reports.

Create Anomaly Reports. Allows a user to create Anomaly Reports for use by your organization. Think of this as an admin-type permission.

To set Anomaly Permissions for roles in your account:

1. Choose **Manage > Users**.
2. Click **View Roles**.
3. Click **Edit** to the right of the role for which you'd like to assign or disable the Anomaly Permissions.
4. Select or deselect **View Anomaly Reports** and **Create Anomaly Reports**.



The screenshot shows a form for configuring a role. It includes fields for 'Name' (containing 'Administrator'), 'Description', and 'Query String'. Below these are two sections of permissions: 'Collector Permissions' and 'Anomaly Permissions'. The 'Anomaly Permissions' section is highlighted with a red box and contains two checked items: 'View Anomaly Reports' and 'Create Anomaly Reports'.

Collector Permissions:	Anomaly Permissions:
<input checked="" type="checkbox"/> Read Collectors	<input checked="" type="checkbox"/> View Anomaly Reports
<input checked="" type="checkbox"/> Manage Collectors	<input checked="" type="checkbox"/> Create Anomaly Reports

5. Click **Save**.

The permissions are applied the next time each user logs in to Sumo Logic.

CHAPTER C

Sumo Logic Beta Features

Want a sneak peek at the latest and greatest Sumo Logic features? You're in the right place!

About Field Extraction

Field Extraction allows you to set up rules that parse out fields as log messages are ingested. This means that instead of running a query to parse out fields, that work is done automatically—when it's time to run a search the fields are parsed and ready to return results.

As data is ingested, Field Extraction Rules are applied to the raw messages. Fields that match a rule are indexed according to the one (or many) applied.

Once it's time to run a query, instead of typing parse commands over and over again, you'll just search against a rule and fields are almost immediately returned.

Creating a Field Extraction Rule

A Field Extraction Rule consists of a query broken into three different text boxes:

- **Rule Name.** The Rule Name is what you'll use later to run searches with the extracted fields. Think of the Rule Name as a metadata element that can be included in queries.
- **Scope.** The Scope of a rule is used to specify the `_sourceCategory`, `_sourceHost`, or `_sourceName` that you'd like to use when fields are parsed. No other metadata values can be used in the Scope field.
- **Fields.** The Fields text box is where the fields you'd like to parse are defined. You can choose to parse one or more fields. Because fields are associated with the Rule Name, you can parse one particular field into as many rules as you'd like. For example, to parse a single field, the definition could look like this: `parse "message count = *," as msg_count`. To parse multiple fields, the definition could look more like this: `parse "[hostId=*" [module=*" [localUserName=*" [logger=*" [thread=*" as hostId, module, localUserName, logger, thread`

What happens if no expressions match a rule?

If no messages match a rule, no fields are parsed or added, meaning that no search results will be returned for that particular rule.

Creating a new Field Extraction Rule

Field Extraction Rules are created and managed using the Field page in the Sumo Logic Web Application. Admins can create their own rules, and delete rules created by other admins.

To create a new Field Extraction Rule:

1. In the Sumo Logic Web Application, choose **Manage > Field Extraction**.
2. Click **Create**.
3. Enter text for Rule Name, then type the scope of the rule as well as the fields you'd like to parse.
4. Click **Add**.

Running a Search Against Extracted Fields

When running a search against Extracted Fields, you'll need to include the exact name of the Rule used to established the extractions.

To run a search against Extracted Fields:

1. Make sure you have the name of the Rule. You can copy the Rule from the Extracted Fields page.
2. In the search box, type or paste the name of the Rule. Optionally, if you'd like, add any operators or other search functions.
3. Start the search.

Using Lookup to Access Saved Data

Once you've saved the results of a search to the Sumo Logic file system using a Save operator, the lookup operator allows you to search that data.

For example, say we wanted to find the date when users signed up in a file named newDailyUsers (the full path is myFolder/mySubFolder/newDailyUsers). We'd use this query to find that information:

```
* | parse "user_name=*" as name  
| lookup date from myFolder/mySubFolder/newDailyUsers on name=name
```